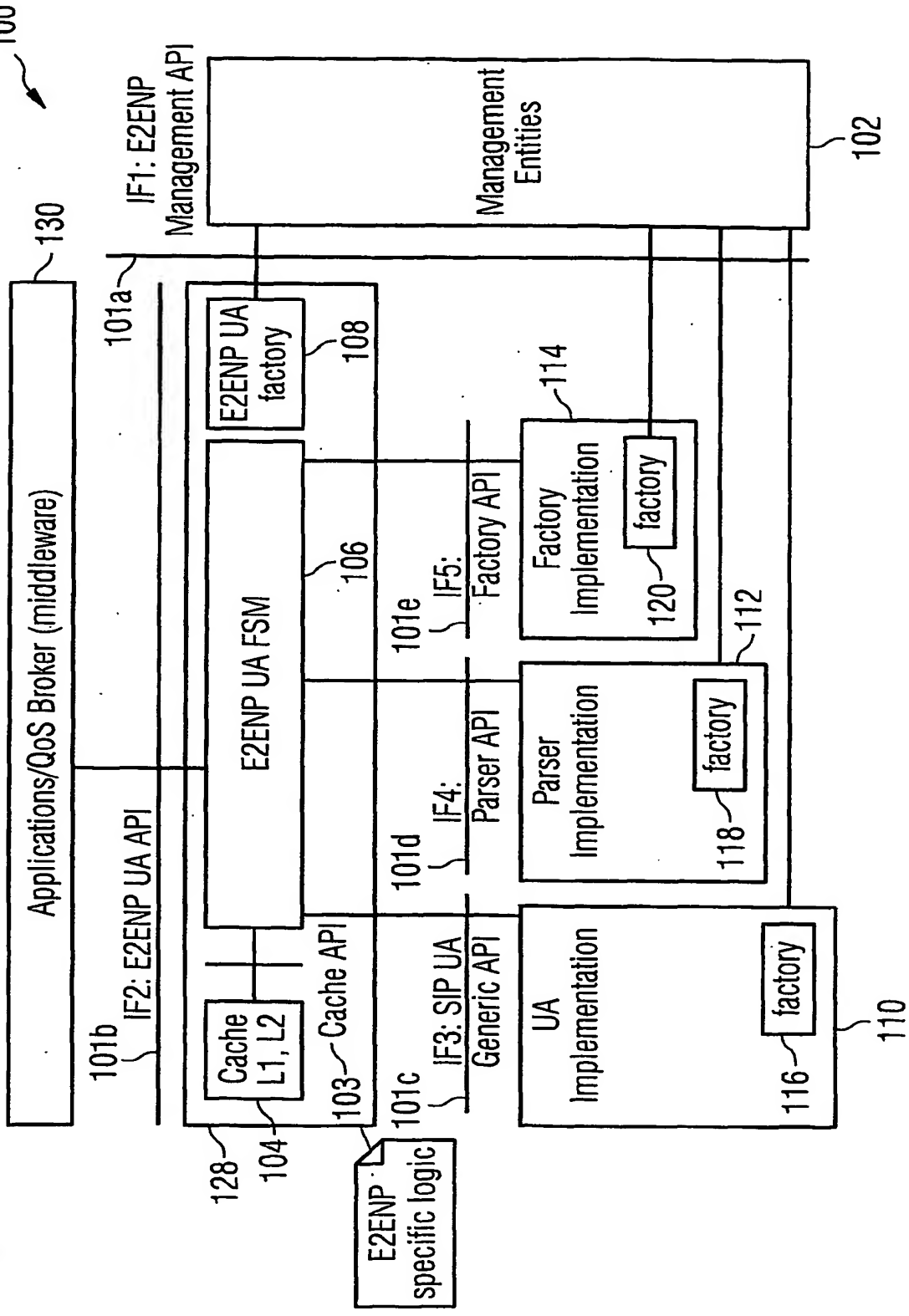

100
Applications/QoS Broker (middleware)
130
IF1: E2ENP Management API
Management Entities
102
101a
101b
IF2: E2ENP UA API
E2ENP UA factory
108
E2ENP UA FSM
106
Cache L1, L2
104
Cache API
103
128
101c
IF3: SIP UA Generic API
101d
IF4: Parser API
101e
IF5: Factory API
E2ENP specific logic
UA Implementation
factory
116
110
Parser Implementation
factory
118
112
Factory Implementation
factory
120
114

FIG 1

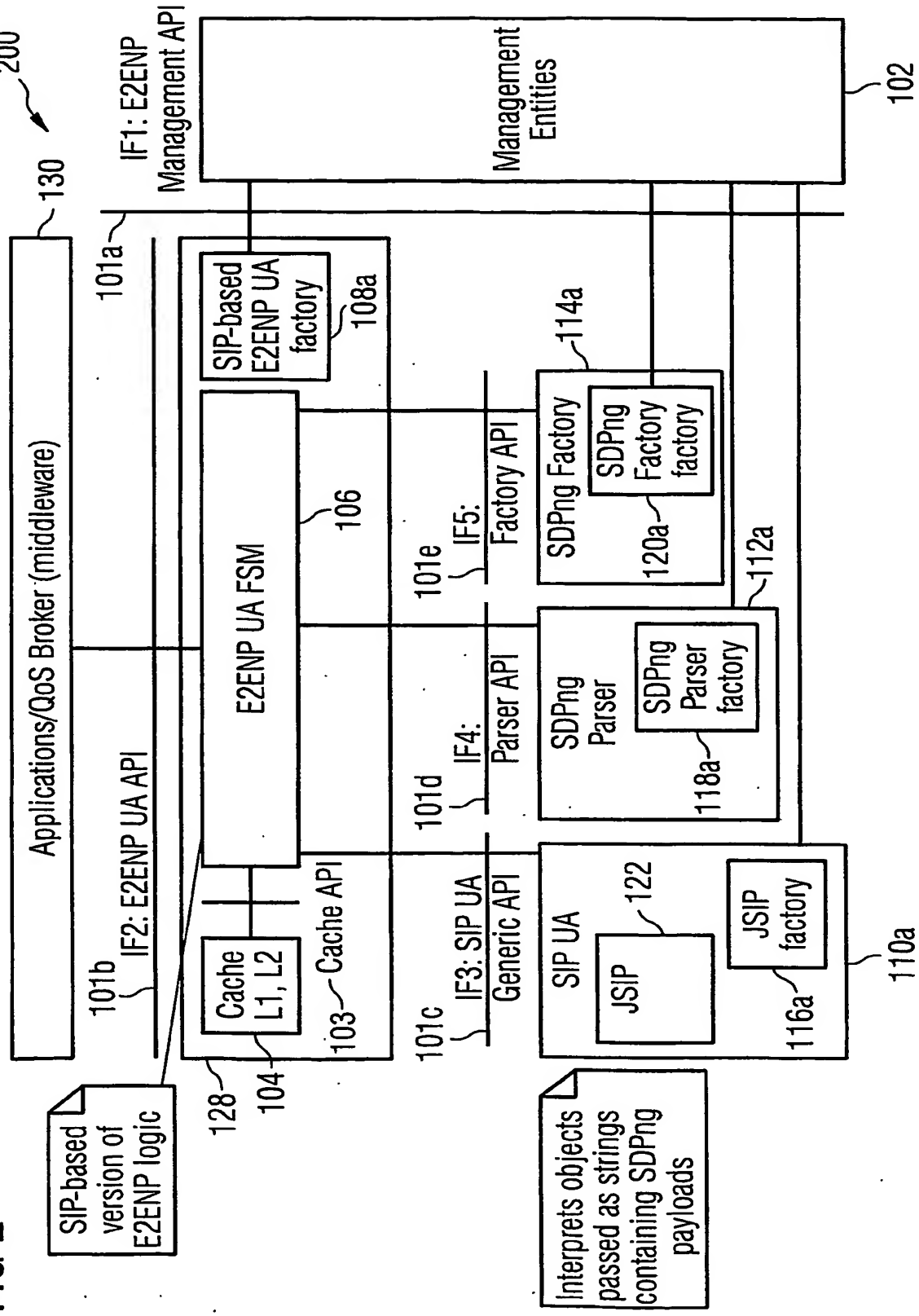

200
130
Applications/QoS Broker (middleware)
101b
IF2: E2ENP UA API
IF1: E2ENP Management API
101a
Management Entities
102
SIP-based E2ENP UA factory
108a
E2ENP UA FSM
106
Cache L1, L2
Cache API
103
128
104
SIP-based version of E2ENP logic
101e
IF5: Factory API
SDPng Factory
SDPng Factory factory
114a
120a
112a
101d
IF4: Parser API
SDPng Parser
SDPng Parser factory
118a
101c
IF3: SIP UA Generic API
SIP UA
JSIP
122
JSIP factory
116a
110a
Interprets objects passed as strings containing SDPng payloads

FIG 2

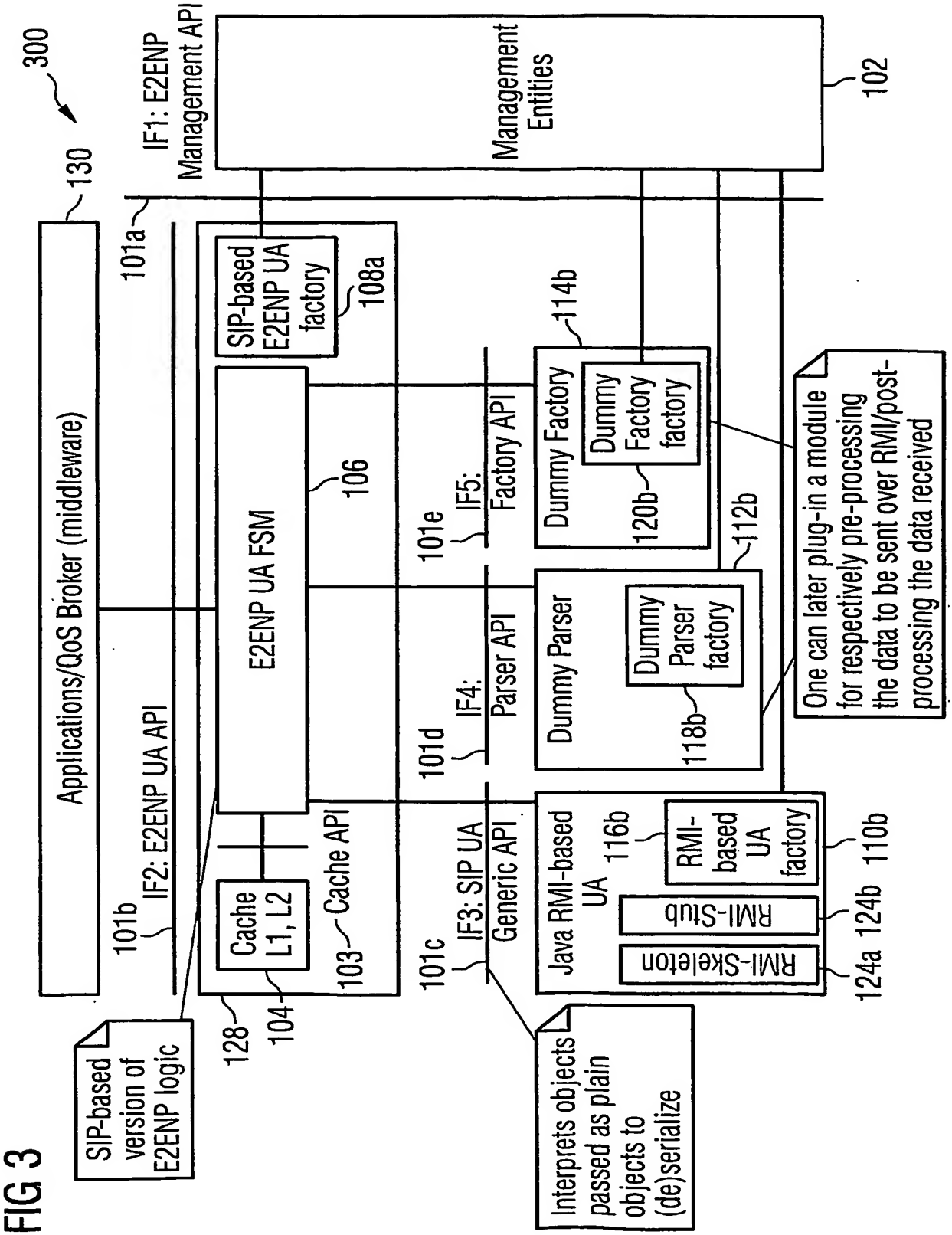

300
130
IF1: E2ENP Management API
Management Entities
102
101a
Applications/QoS Broker (middleware)
IF2: E2ENP UA API
101b
SIP-based E2ENP UA factory
108a
E2ENP UA FSM
106
Cache L1, L2
Cache API
103
104
128
SIP-based version of E2ENP logic
101e
IF5: Factory API
Dummy Factory
Dummy Factory factory
114b
120b
112b
101d
IF4: Parser API
Dummy Parser
Dummy Parser factory
118b
One can later plug-in a module for respectively pre-processing the data to be sent over RMI/post-processing the data received
101c
IF3: SIP UA Generic API
Java RMI-based UA
116b
RMI-based UA factory
110b
RMI-Stub
RMI-Skeleton
124a
124b
Interprets objects passed as plain objects to (de)serialize

FIG 3

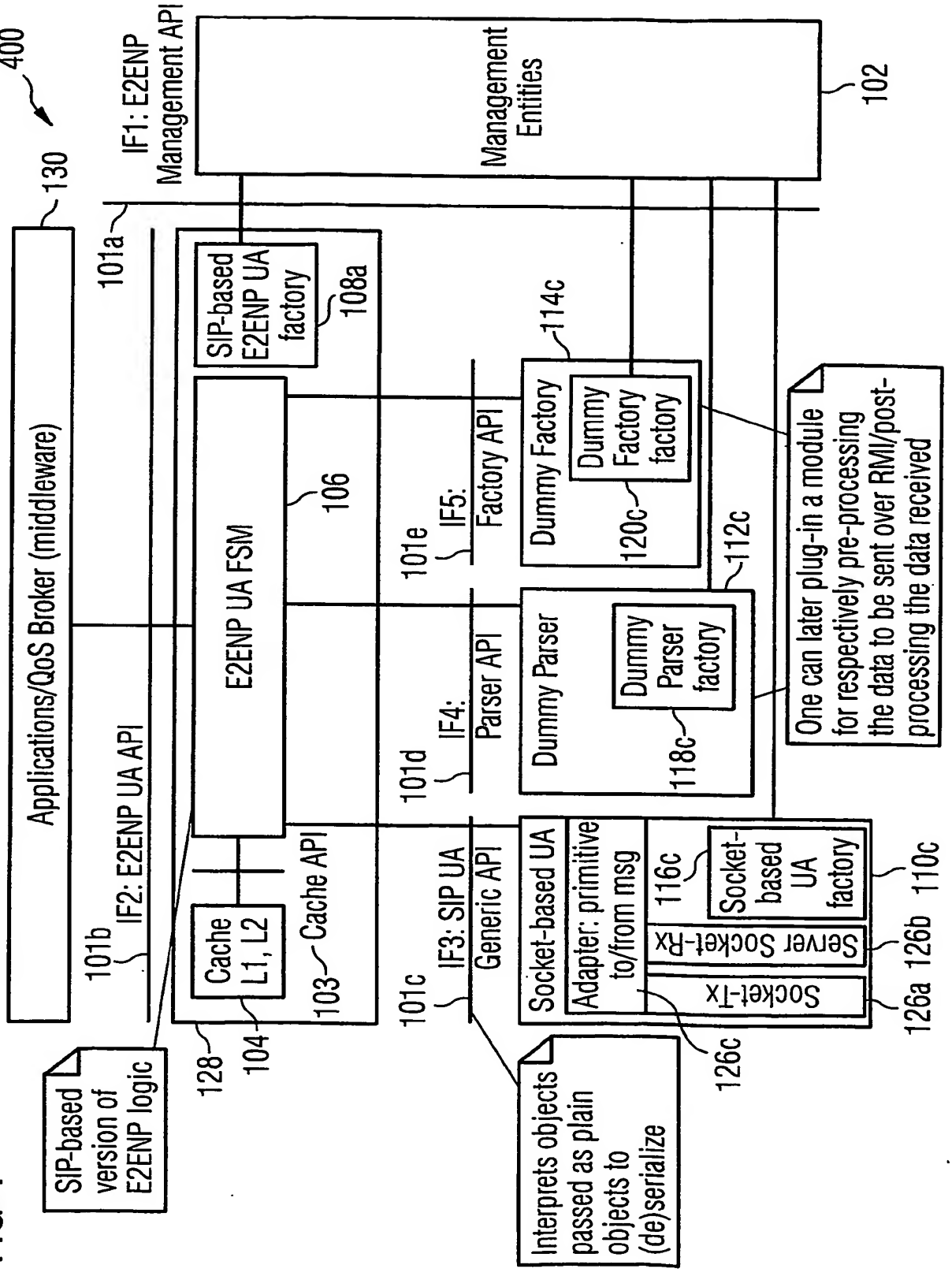

400
130
IF1: E2ENP Management API
Management Entities
102
101a
Applications/QoS Broker (middleware)
SIP-based E2ENP UA factory
108a
E2ENP UA FSM
106
IF5: Factory API
Dummy Factory
Dummy Factory factory
114c
120c
112c
101e
101b
IF2: E2ENP UA API
Cache L1, L2
Cache API
103
104
128
101d
IF4: Parser API
Dummy Parser
Dummy Parser factory
118c
One can later plug-in a module for respectively pre-processing the data to be sent over RMI/post-processing the data received
101c
IF3: SIP UA Generic API
Socket-based UA
Adapter: primitive to/from msg
116c
Socket-based UA factory
110c
Server Socket-Rx
Socket-Tx
126a
126b
126c
SIP-based version of E2ENP logic
Interprets objects passed as plain objects to (de)serialize

FIG 4

FIG 5

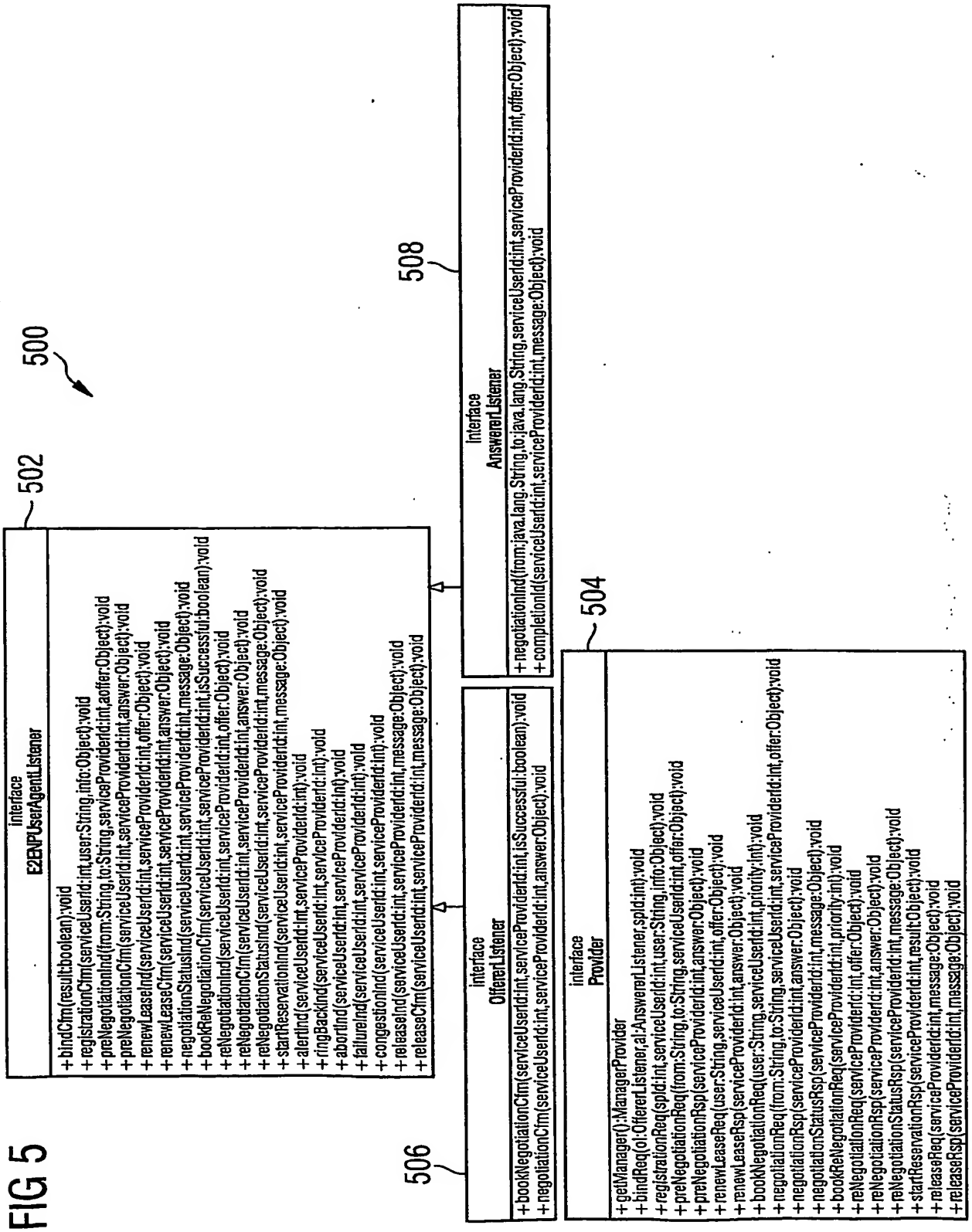

500
502
interface
E2EPUserAgentListener
+bindCfm(result:boolean):void
+registrationCfm(serviceUserId:int,user:String,info:Object):void
+preNegotiationInd(from:String,to:String,serviceProviderId:int,aoffer:Object):void
+preNegotiationCfm(serviceUserId:int,serviceProviderId:int,answer:Object):void
+renewLeaseInd(serviceUserId:int,serviceProviderId:int,offer:Object):void
+renewLeaseCfm(serviceUserId:int,serviceProviderId:int,answer:Object):void
+negotiationStatusInd(serviceUserId:int,serviceProviderId:int,message:Object):void
+bookReNegotiationCfm(serviceUserId:int,serviceProviderId:int,isSuccessful:boolean):void
+reNegotiationInd(serviceUserId:int,serviceProviderId:int,offer:Object):void
+reNegotiationCfm(serviceUserId:int,serviceProviderId:int,answer:Object):void
+reNegotiationStatusInd(serviceUserId:int,serviceProviderId:int,message:Object):void
+startReservationInd(serviceUserId:int,serviceProviderId:int,message:Object):void
+alertInd(serviceUserId:int,serviceProviderId:int):void
+ringBackInd(serviceUserId:int,serviceProviderId:int):void
+abortInd(serviceUserId:int,serviceProviderId:int):void
+failureInd(serviceUserId:int,serviceProviderId:int):void
+congestionInd(serviceUserId:int,serviceProviderId:int):void
+releaseInd(serviceUserId:int,serviceProviderId:int,message:Object):void
+releaseCfm(serviceUserId:int,serviceProviderId:int,message:Object):void
506
interface
OffererListener
+bookNegotiationCfm(serviceUserId:int,serviceProviderId:int,isSuccessful:boolean):void
+negotiationCfm(serviceUserId:int,serviceProviderId:int,answer:Object):void
508
interface
AnswererListener
+negotiationInd(from:java.lang.String,to:java.lang.String,serviceUserId:int,serviceProviderId:int,offer:Object):void
+completionId(serviceUserId:int,serviceProviderId:int,message:Object):void
504
interface
Provider
+getManager():ManagerProvider
+bindReq(ol:OffererListener,al:AnswererListener,spId:int):void
+registrationReq(spId:int,serviceUserId:int,user:String,info:Object):void
+preNegotiationReq(from:String,to:String,serviceUserId:int,offer:Object):void
+preNegotiationRsp(serviceProviderId:int,answer:Object):void
+renewLeaseReq(user:String,serviceUserId:int,offer:Object):void
+renewLeaseRsp(serviceProviderId:int,answer:Object):void
+bookNegotiationReq(user:String,serviceUserId:int,priority:int):void
+negotiationReq(from:String,to:String,serviceUserId:int,serviceProviderId:int,offer:Object):void
+negotiationRsp(serviceProviderId:int,answer:Object):void
+negotiationStatusRsp(serviceProviderId:int,message:Object):void
+bookReNegotiationReq(serviceProviderId:int,priority:int):void
+reNegotiationReq(serviceProviderId:int,offer:Object):void
+reNegotiationRsp(serviceProviderId:int,answer:Object):void
+reNegotiationStatusRsp(serviceProviderId:int,message:Object):void
+startReservationRsp(serviceProviderId:int,result:Object):void
+releaseReq(serviceProviderId:int,message:Object):void
+releaseRsp(serviceProviderId:int,message:Object):void

FIG 5bis

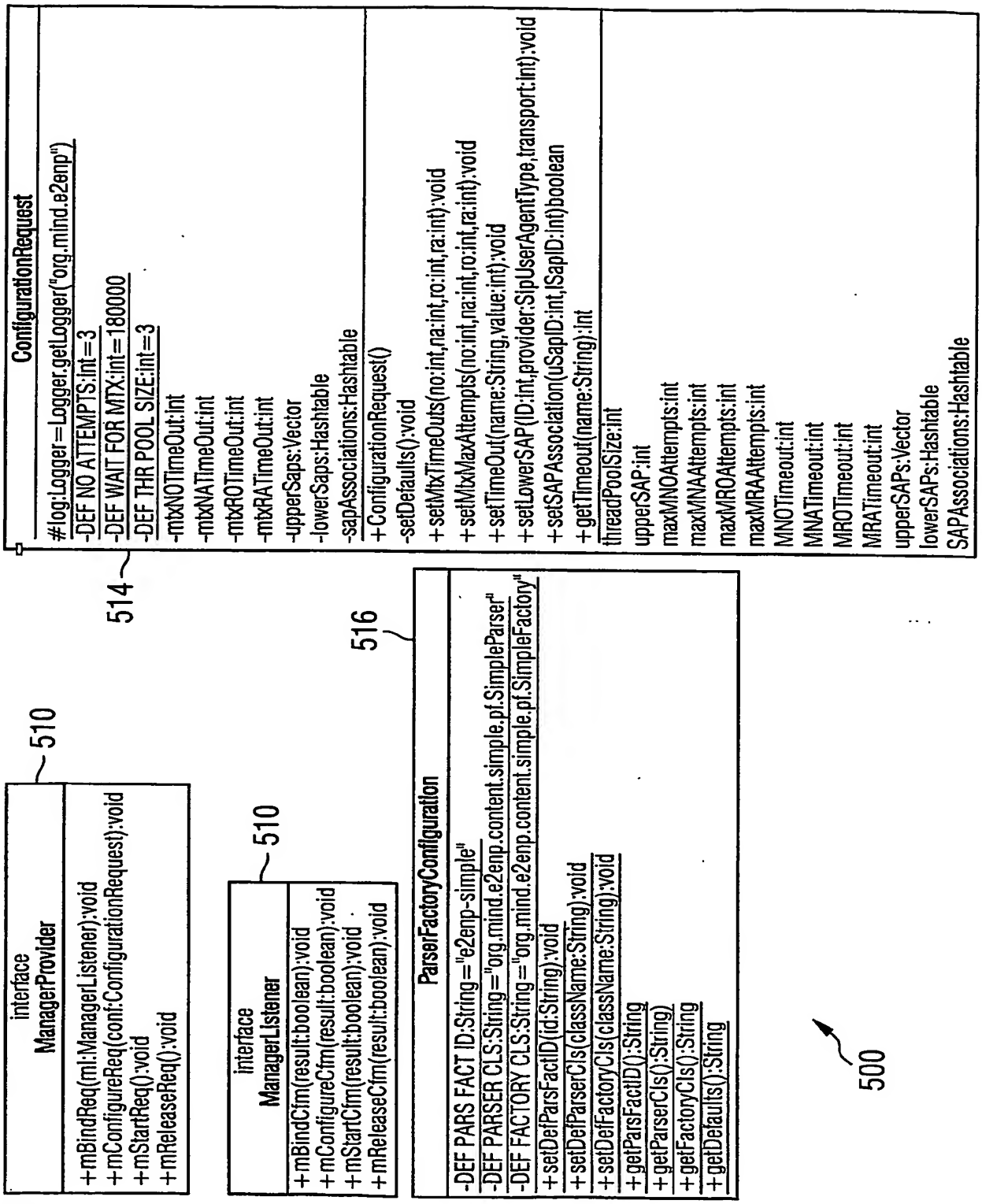

interface
ManagerProvider
+mBindReq(ml:ManagerListener):void
+mConfigureReq(conf:ConfigurationRequest):void
+mStartReq():void
+mReleaseReq():void
510
interface
ManagerListener
+mBindCfm(result:boolean):void
+mConfigureCfm(result:boolean):void
+mStartCfm(result:boolean):void
+mReleaseCfm(result:boolean):void
510
ParserFactoryConfiguration
-DEF PARS FACT ID:String="e2enp-simple"
-DEF PARSER CLS:String="org.mind.e2enp.content.simple.pf.SimpleParser"
-DEF FACTORY CLS:String="org.mind.e2enp.content.simple.pf.SimpleFactory"
+setDefParsFactID(id:String):void
+setDefParserCls(className:String):void
+setDefFactoryCls(className:String):void
+getParsFactID():String
+getParserCls():String)
+getFactoryCls():String
+getDefaults():String
516
ConfigurationRequest
#log:Logger=Logger.getLogger("org.mind.e2enp")
-DEF NO ATTEMPTS:int=3
-DEF WAIT FOR MTX:int=180000
-DEF THR POOL SIZE:int=3
-mtxNOTimeOut:int
-mtxNATimeOut:int
-mtxROTimeOut:int
-mtxRATimeOut:int
-upperSaps:Vector
-lowerSaps:Hashtable
-sapAssociations:Hashtable
+ConfigurationRequest()
-setDefaults():void
+setMtxTimeOuts(no:int,na:int,ro:int,ra:int):void
+setMtxMaxAttempts(no:int,na:int,ro:int,ra:int):void
+setTimeOut(name:String,value:int):void
+setLowerSAP(ID:int,provider:SipUserAgentType,transport:int):void
+setSAPAssociation(uSapID:int,lSapID:int)boolean
+getTimeout(name:String):int
threadPoolSize:int
upperSAP:int
maxMNOAttempts:int
maxMNAAttempts:int
maxMROAttempts:int
maxMRAAttempts:int
MNOTimeout:int
MNATimeout:int
MROTimeout:int
MRATimeout:int
upperSAPs:Vector
lowerSAPs:Hashtable
SAPAssociations:Hashtable
514
500

FIG 6

E2ENP-URI	=	"e2enp:// [userinfo "@"] hostport
userinfo	=	[user] [":" password]
user-unreserved	=	"&" / "=" / "+" / "$" / "," / ";"/ "?"
escaped	=	"%" HEXDIG HEXDIG
unreserved	=	alphanum / mark
mark	=	"-" / "_" / "." / "!" / "~" / "*" / "'" / "(" / ")"
alphanum	=	ALPHA / DIGIT
user	=	*(unreserved / escaped / user-unreserved)
password	=	*(unreserved / escaped / "&" / "=" / "+" / "$" / ",")
hostport	=	host / host ":" port
host	=	hostname / hostaddress
hostname	=	*(domainlabel ".") toplabel ["."]
domainlabel	=	alphanum / alphanum * (alphanum / "-") alphanum
toplabel	=	ALPHA / ALPHA * (alphanum / "-") alphanum
hostaddress	=	ipv4-addr / ipv6-addr
ipv4-addr	=	digits"."digits"."digits"."digits
digits	=	1*3DIGIT
ipv6-addr	=	hexpart [":" IPv4address]
hexpart	=	hexseq / hexseq "::" [hexseq] / "::" [hexseq]
hexseq	=	hex4 *(":" hex4)
hex4	=	1*4HEXDIG
port	=	1*DIGIT

600

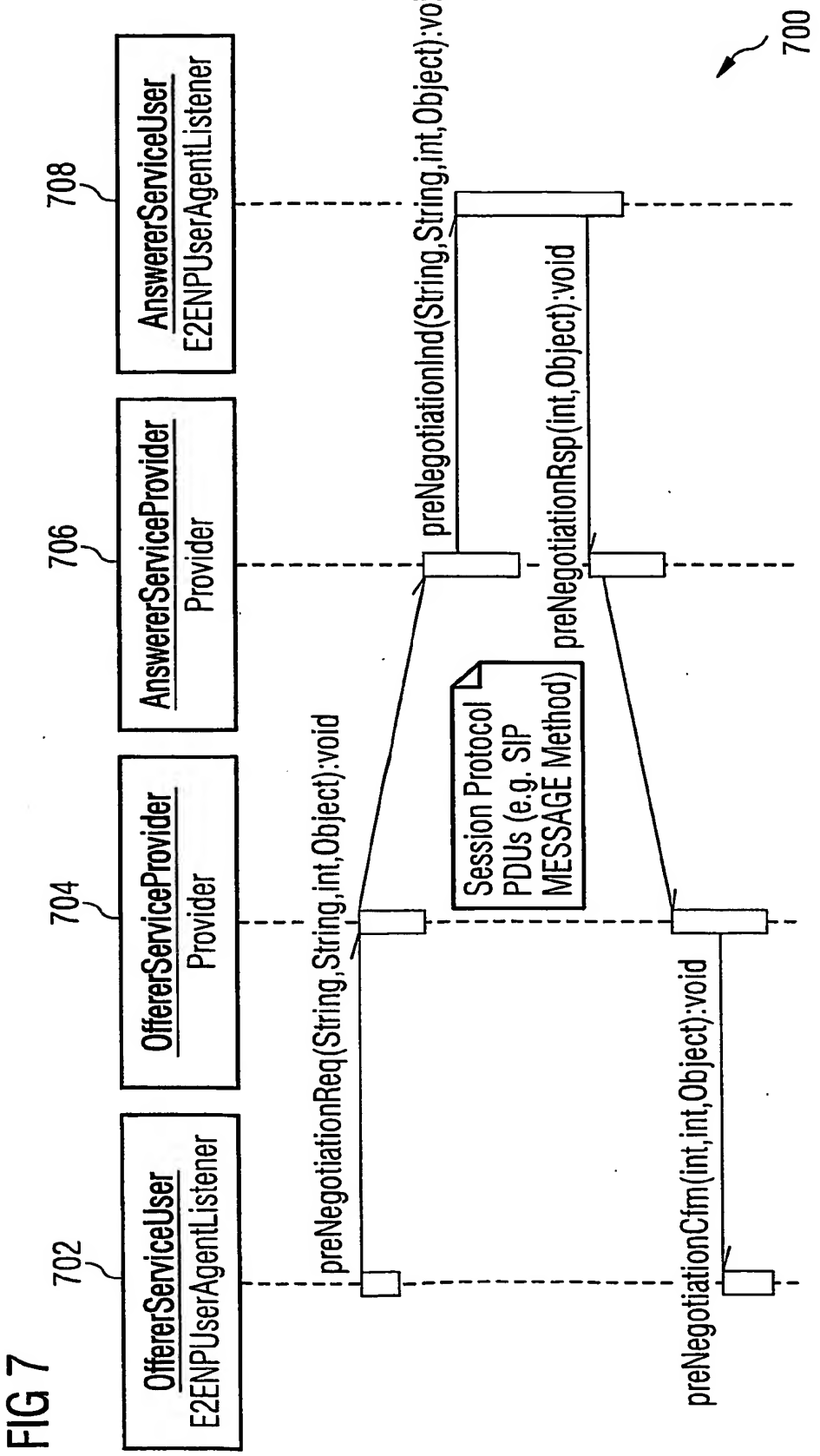
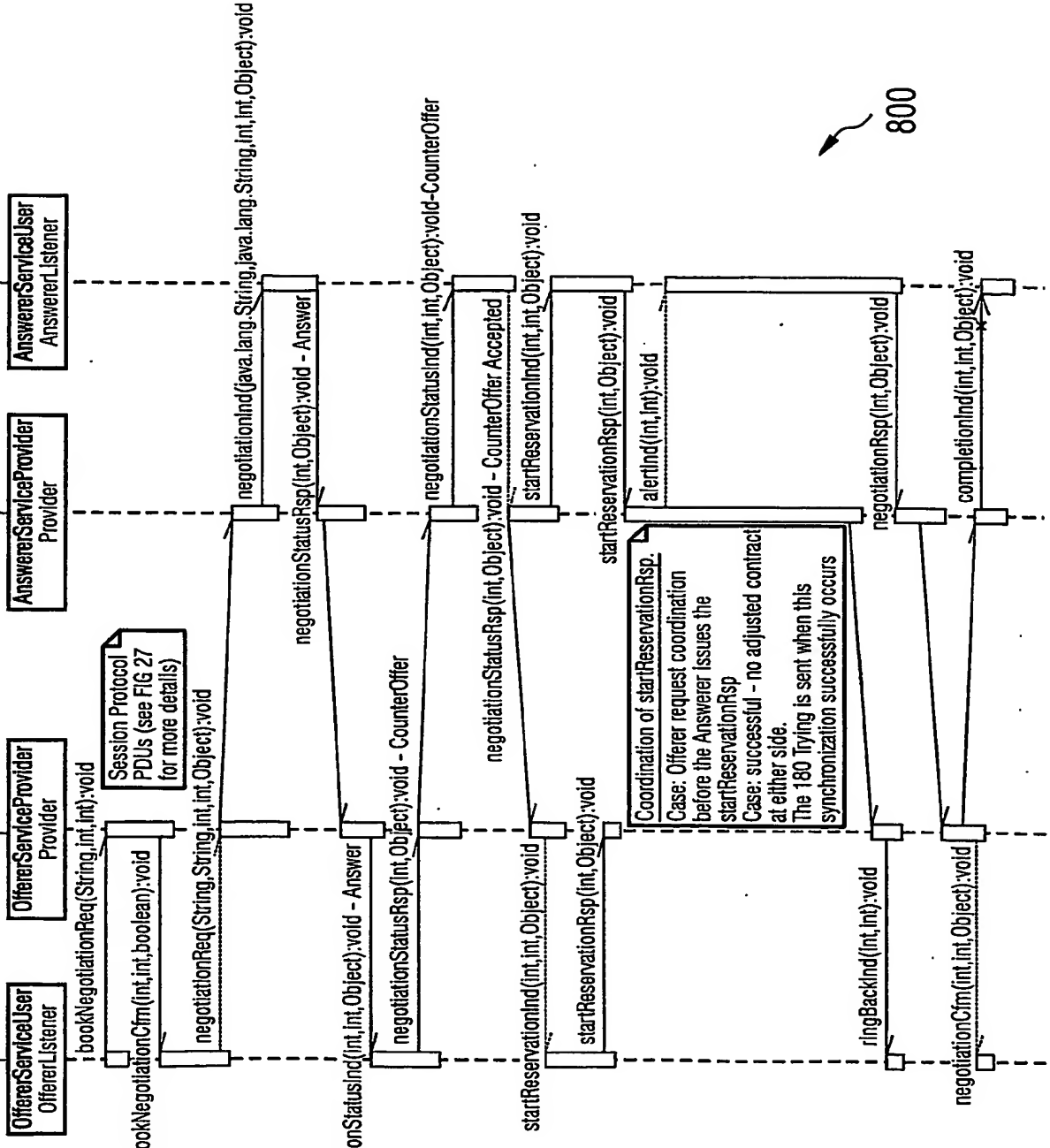

700
702
OfferierServiceUser
E2ENPUserAgentListener
704
OfferierServiceProvider
Provider
706
AnswererServiceProvider
Provider
708
AnswererServiceUser
E2ENPUserAgentListener
preNegotiationReq(String,String,int,Object):void
preNegotiationInd(String,String,int,Object):void
preNegotiationRsp(int,Object):void
preNegotiationCfm(int,int,Object):void
Session Protocol
PDUs (e.g. SIP
MESSAGE Method)

FIG 7

FIG 8


800
702
704
706
708
OffererServiceUser
OffererListener
OffererServiceProvider
ProvIder
AnswererServiceProvider
ProvIder
AnswererServiceUser
AnswererListener
1 bookNegotiationReq(String,int,int):void
bookNegotiationCfm(int,int,boolean):void
negotiationReq(String,String,int,int,Object):void
Session Protocol PDUs (see FIG 27 for more details)
negotiationInd(java.lang.String,java.lang.String,int,int,Object):void
negotiationStatusRsp(int,Object):void - Answer
negotiationStatusInd(int,int,Object):void - Answer
negotiationStatusRsp(int,Object):void - CounterOffer
negotiationStatusInd(int,int,Object):void-CounterOffer
negotiationStatusRsp(int,Object):void - CounterOffer Accepted
startReservationInd(int,int,Object):void
startReservationRsp(int,Object):void
startReservationRsp(int,Object):void
Coordination of startReservationRsp.
Case: Offerer request coordination before the Answerer issues the startReservationRsp
Case: successful - no adjusted contract at either side.
The 180 Trying is sent when this synchronization successfully occurs
alertInd(int,int):void
negotiationRsp(int,Object):void
ringBackInd(int,int):void
completionInd(int,int,Object):void
negotiationCfm(int,int,Object):void

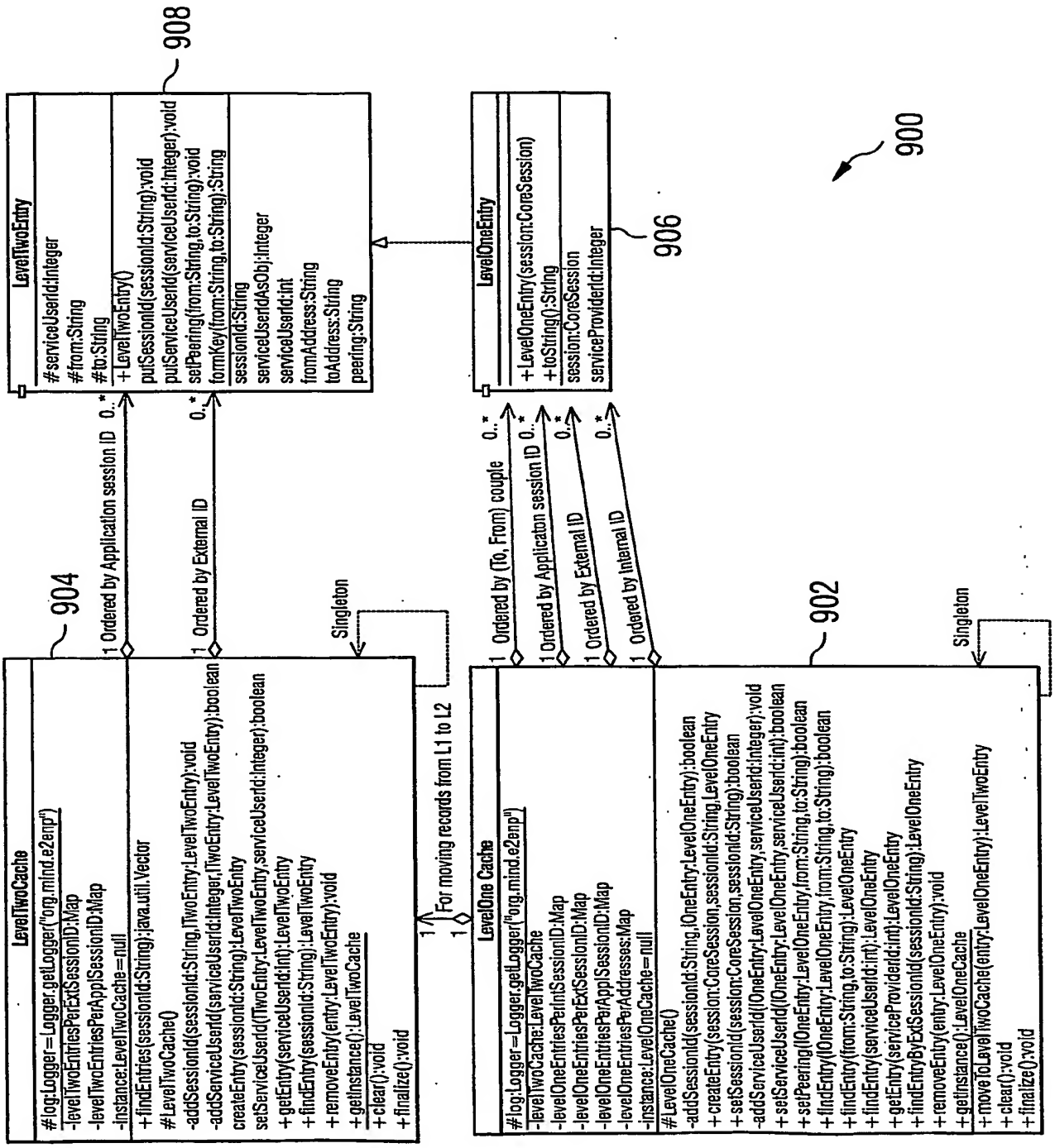

LevelTwoCache
#log:Logger=Logger.getLogger("org.mind.e2enp")
-levelTwoEntriesPerExtSessionID:Map
-levelTwoEntriesPerApplSessionID:Map
-instance:LevelTwoCache=null
+findEntries(sessionId:String):java.util.Vector
#LevelTwoCache()
-addSessionId(sessionId:String,lTwoEntry:LevelTwoEntry):void
-addServiceUserId(serviceUserId:Integer,lTwoEntry:LevelTwoEntry):boolean
createEntry(sessionId:String):LevelTwoEntry
setServiceUserId(lTwoEntry:LevelTwoEntry,serviceUserId:Integer):boolean
+getEntry(serviceUserId:int):LevelTwoEntry
+findEntry(sessionId:String):LevelTwoEntry
+removeEntry(entry:LevelTwoEntry):void
+getInstance():LevelTwoCache
+clear():void
+finalize():void
904
1 Ordered by Application session ID 0..*
1 Ordered by External ID 0..*
Singleton
For moving records from L1 to L2
1
1
LevelTwoEntry
#serviceUserId:Integer
#from:String
#to:String
+LevelTwoEntry()
putSessionId(sessionId:String):void
putServiceUserId(serviceUserId:Integer):void
setPeering(from:String,to:String):void
formKey(from:String,to:String):String
sessionId:String
serviceUserIdAsObj:Integer
serviceUserId:int
fromAddress:String
toAddress:String
peering:String
908
1 Ordered by (To, From) couple 0..*
1 Ordered by Applicaton session ID 0..*
1 Ordered by External ID 0..*
1 Ordered by Internal ID 0..*
LevelOneEntry
+LevelOneEntry(session:CoreSession)
+toString():String
session:CoreSession
serviceProviderId:Integer
906
900
LevelOne Cache
#log:Logger=Logger.getLogger("org.mind.e2enp")
-levelTwoCache:LevelTwoCache
-levelOneEntriesPerIntSessionID:Map
-levelOneEntriesPerExtSessionID:Map
-levelOneEntriesPerApplSessionID:Map
-levelOneEntriesPerAddresses:Map
-instance:LevelOneCache=null
#LevelOneCache()
-addSessionId(sessionId:String,lOneEntry:LevelOneEntry):boolean
+createEntry(session:CoreSession,sessionId:String,LevelOneEntry
+setSessionId(session:CoreSession,sessionId:String):boolean
-addServiceUserId(lOneEntry:LevelOneEntry,serviceUserId:Integer):void
+setServiceUserId(lOneEntry:LevelOneEntry,serviceUserId:int):boolean
+setPeering(lOneEntry:LevelOneEntry,from:String,to:String):boolean
+findEntry(lOneEntry:LevelOneEntry,from:String,to:String):boolean
+findEntry(from:String,to:String):LevelOneEntry
+findEntry(serviceUserId:int):LevelOneEntry
+getEntry(serviceProviderId:int):LevelOneEntry
+findEntryByExtSessionId(sessionId:String):LevelOneEntry
+removeEntry(entry:LevelOneEntry):void
+getInstance():LevelOneCache
+moveToLevelTwoCache(entry:LevelOneEntry):LevelTwoEntry
+clear():void
+finalize():void
902
Singleton

FIG 9

FIG 10

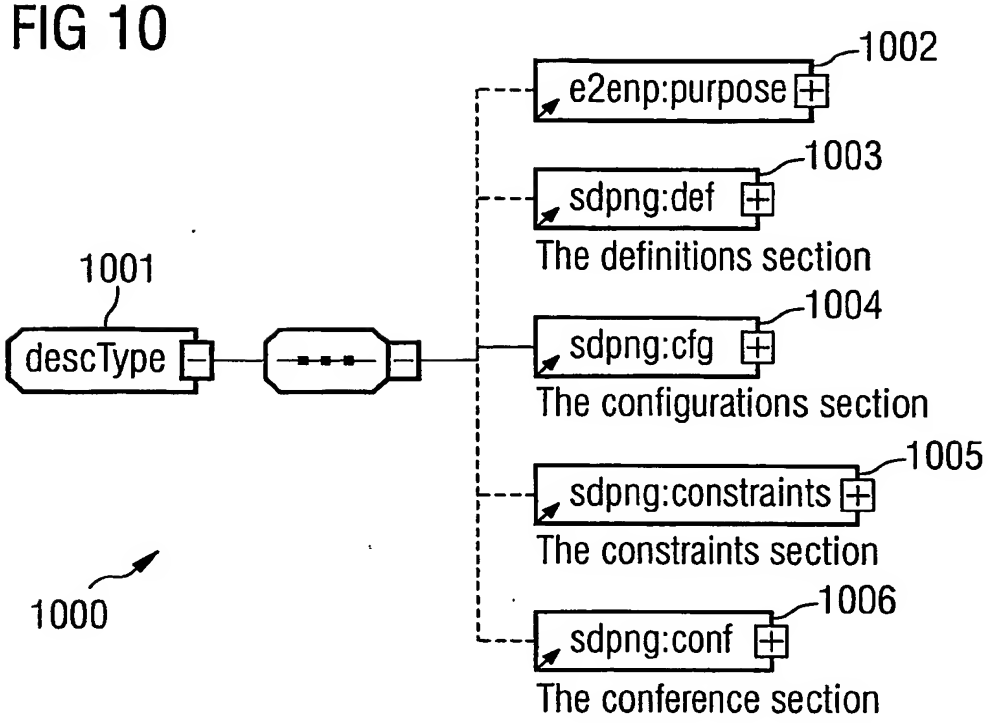

1002
e2enp:purpose
1003
sdpng:def
The definitions section
1001
descType
1004
sdpng:cfg
The configurations section
1005
sdpng:constraints
The constraints section
1000
1006
sdpng:conf
The conference section

FIG 11

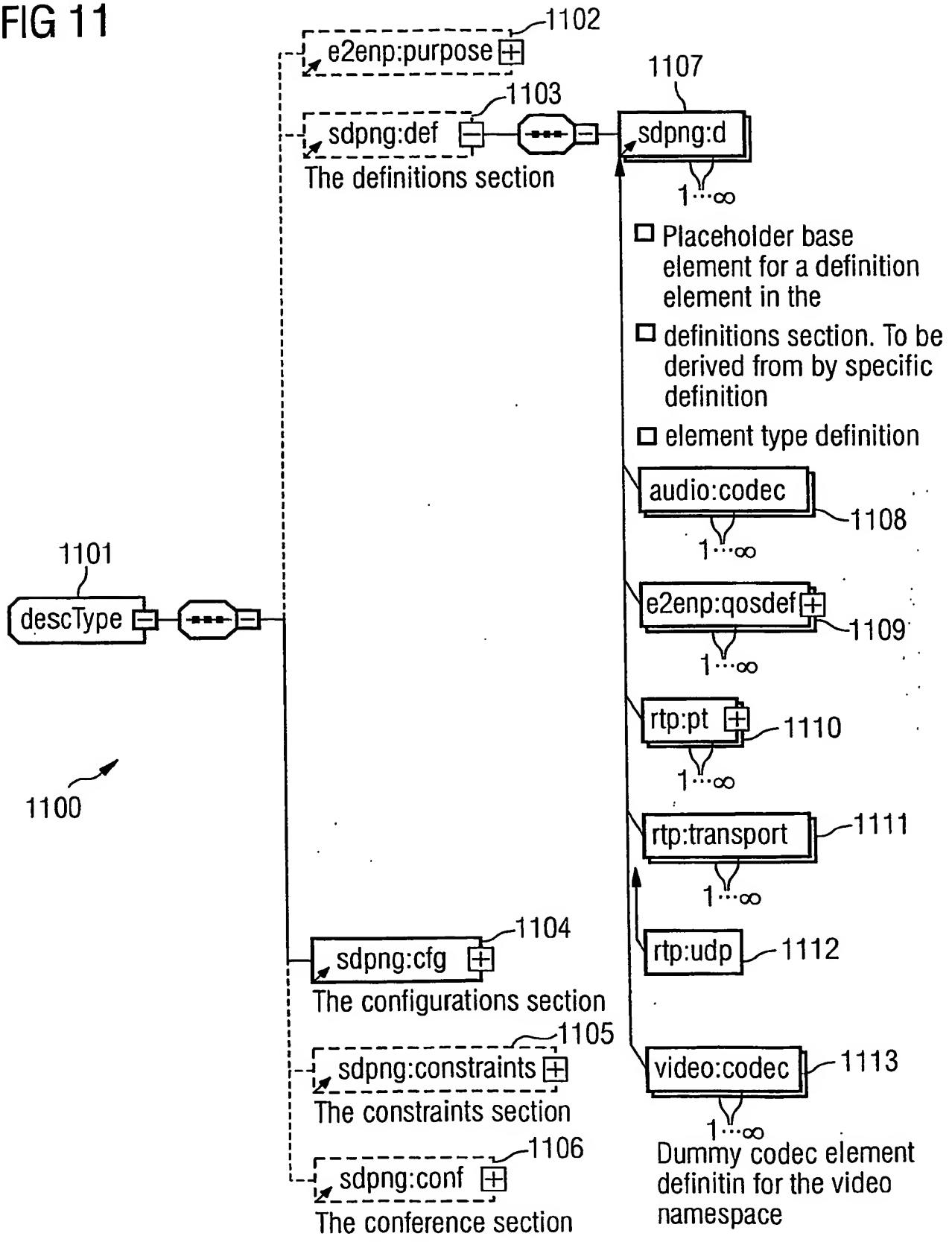

1102
e2enp:purpose
1107
1103
sdpng:def
sdpng:d
The definitions section
1...∞
Placeholder base element for a definition element in the
definitions section. To be derived from by specific definition
element type definition
audio:codec
1108
1...∞
1101
descType
e2enp:qosdef
1109
1...∞
rtp:pt
1110
1...∞
1100
rtp:transport
1111
1...∞
1104
sdpng:cfg
rtp:udp
1112
The configurations section
1105
sdpng:constraints
video:codec
1113
The constraints section
1...∞
1106
Dummy codec element definitin for the video namespace
sdpng:conf
The conference section

FIG 12

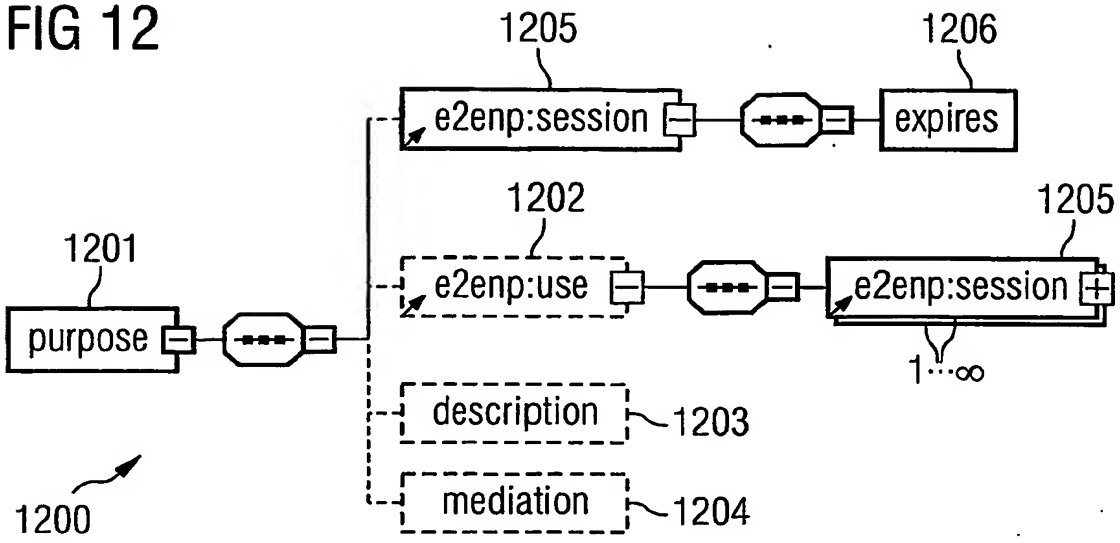

1205
1206
e2enp:session
expires
1202
1205
1201
e2enp:use
e2enp:session
purpose
1...∞
description
1203
mediation
1204
1200

FIG 13

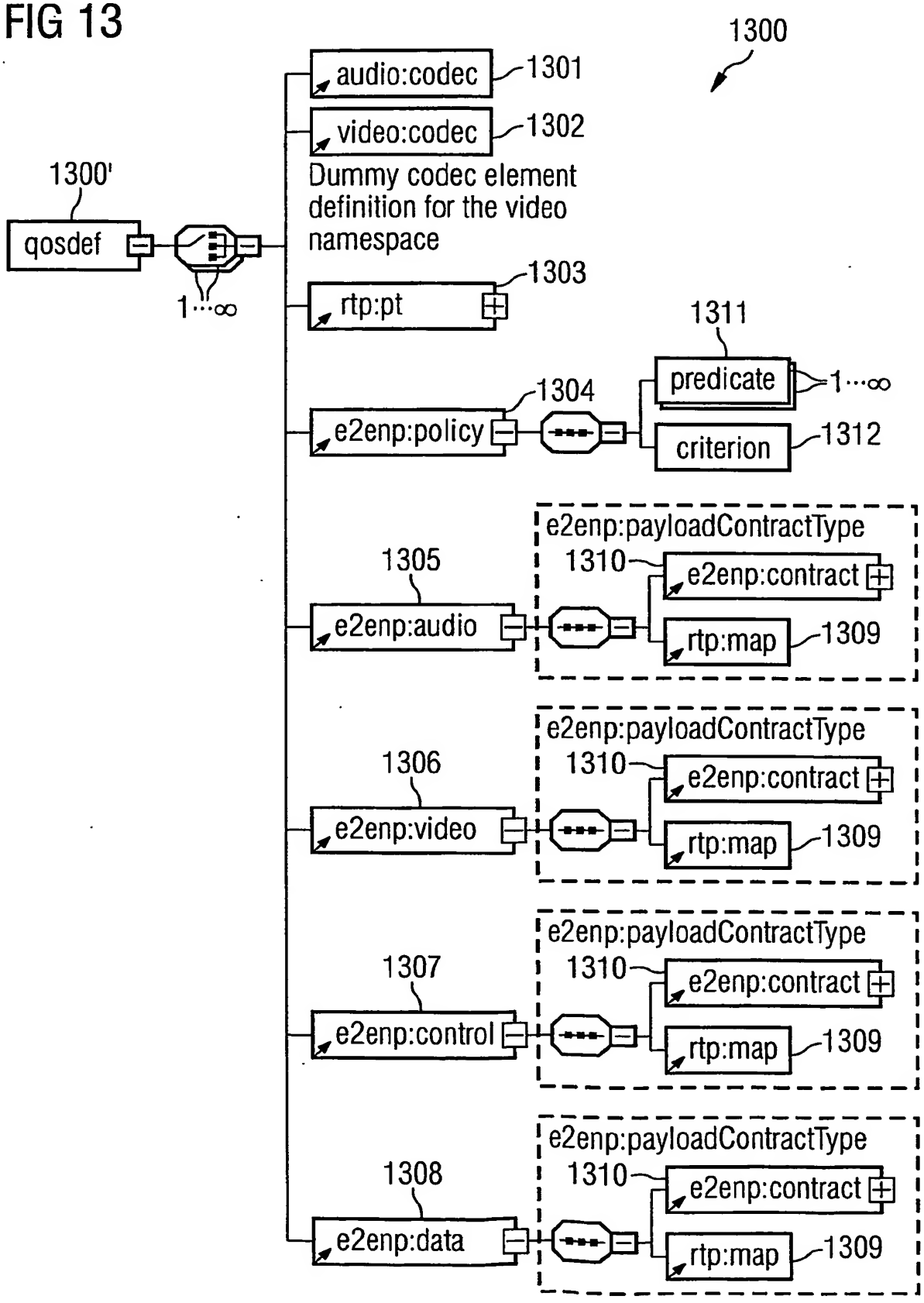
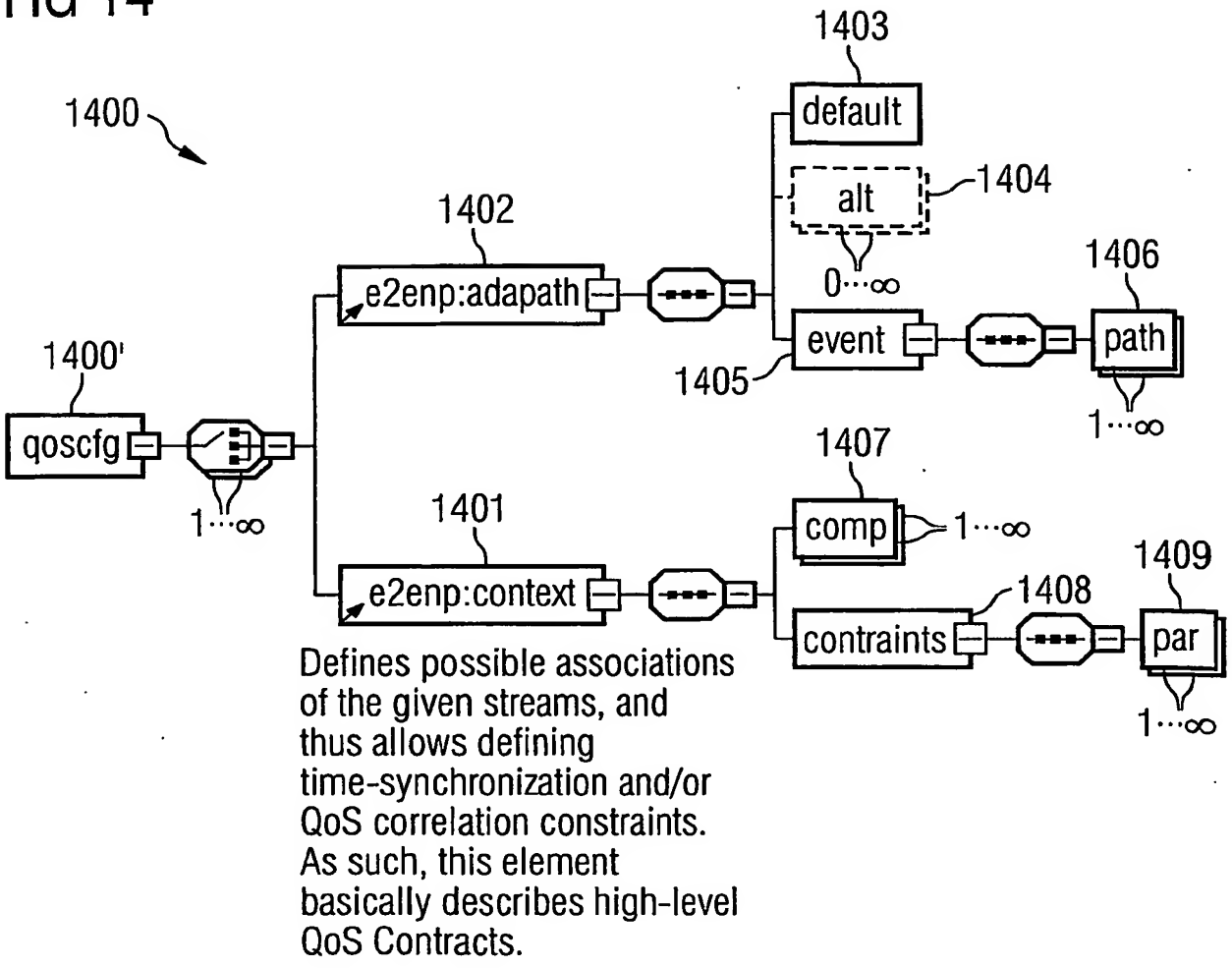

1300
audio:codec
1301
video:codec
1302
1300'
Dummy codec element definition for the video namespace
qosdef
1···∞
1303
rtp:pt
1311
predicate
1···∞
1304
e2enp:policy
criterion
1312
e2enp:payloadContractType
1305
1310
e2enp:contract
e2enp:audio
rtp:map
1309
e2enp:payloadContractType
1306
1310
e2enp:contract
e2enp:video
rtp:map
1309
e2enp:payloadContractType
1307
1310
e2enp:contract
e2enp:control
rtp:map
1309
e2enp:payloadContractType
1308
1310
e2enp:contract
e2enp:data
rtp:map
1309

FIG 14


1400
1403
default
1404
alt
0···∞
1402
e2enp:adapath
1406
1400'
event
path
1405
1···∞
qoscfg
1···∞
1407
1401
comp
1···∞
1409
e2enp:context
1408
contraints
par
1···∞
Defines possible associations of the given streams, and thus allows defining time-synchronization and/or QoS correlation constraints. As such, this element basically describes high-level QoS Contracts.

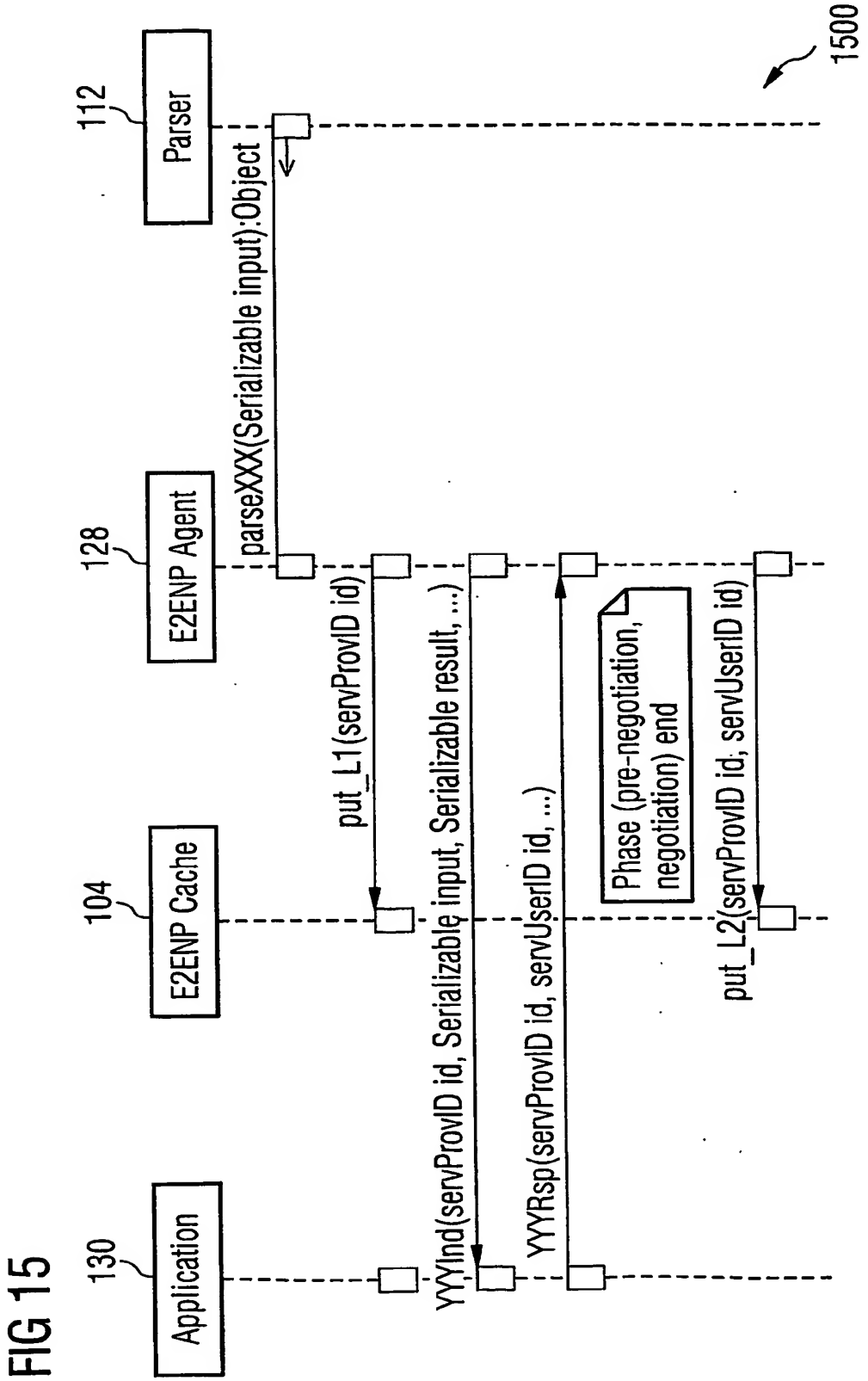

1500
112
Parser
128
E2ENP Agent
104
E2ENP Cache
130
Application
parseXXX(Serializable input):Object
put_L1(servProvID id)
YYYInd(servProvID id, Serializable input, Serializable result, ...)
YYYRsp(servProvID id, servUserID id, ...)
Phase (pre-negotiation, negotiation) end
put_L2(servProvID id, servUserID id)

FIG 15

FIG 17

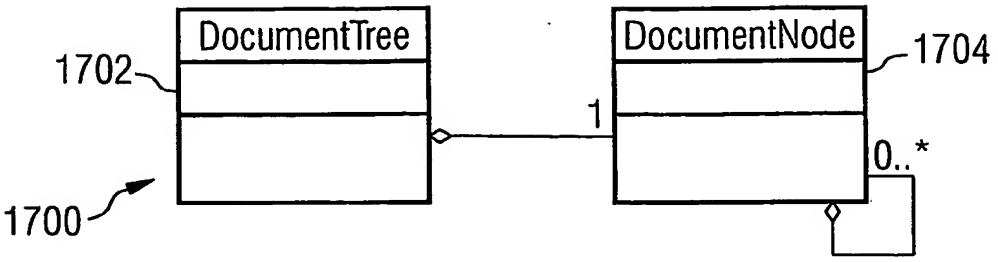
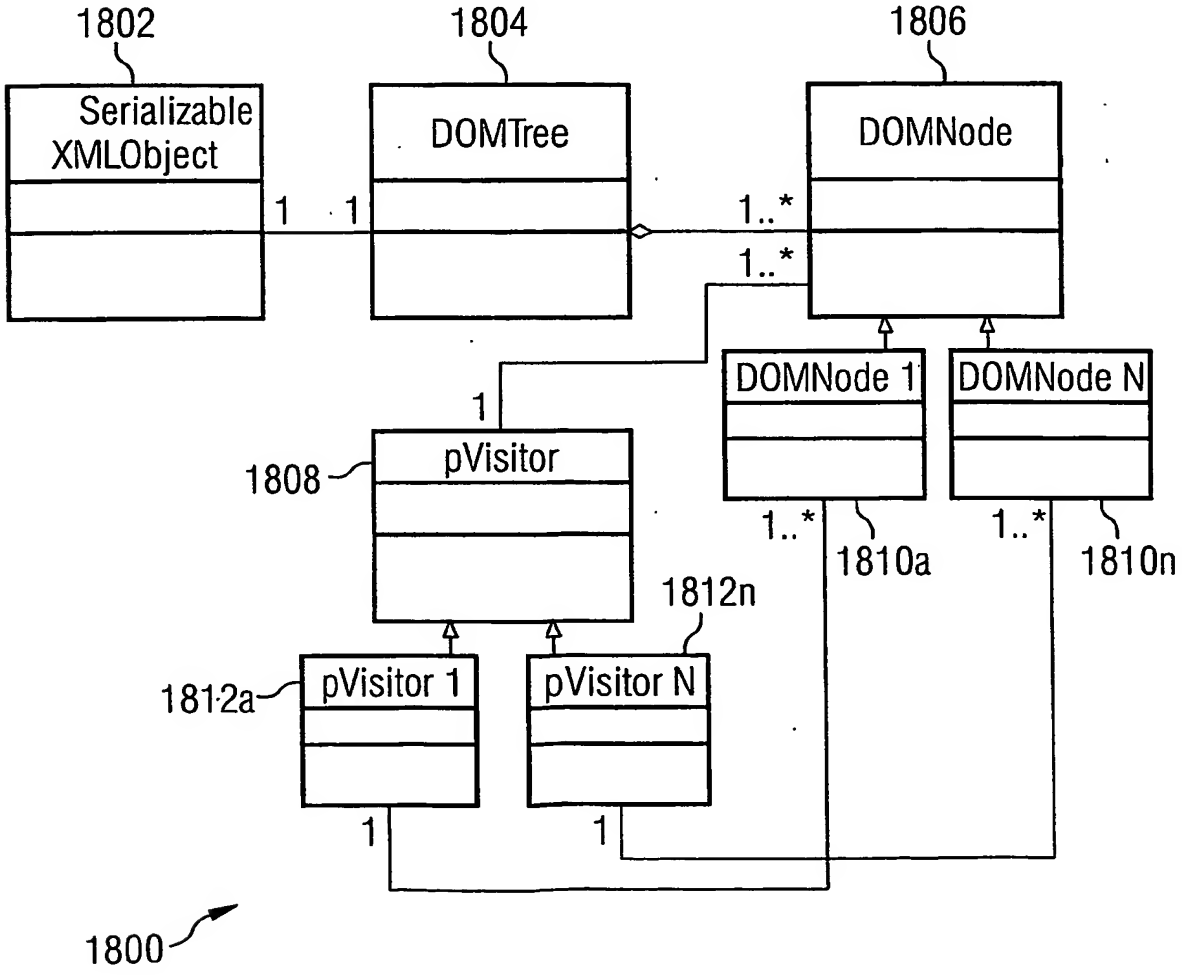

DocumentTree
DocumentNode
1702
1704
1
0..*
1700

FIG 18


1802
1804
1806
Serializable XMLObject
DOMTree
DOMNode
1
1
1..*
1..*
1
DOMNode 1
DOMNode N
1808
pVisitor
1..*
1..*
1810a
1810n
1812n
1812a
pVisitor 1
pVisitor N
1
1
1800

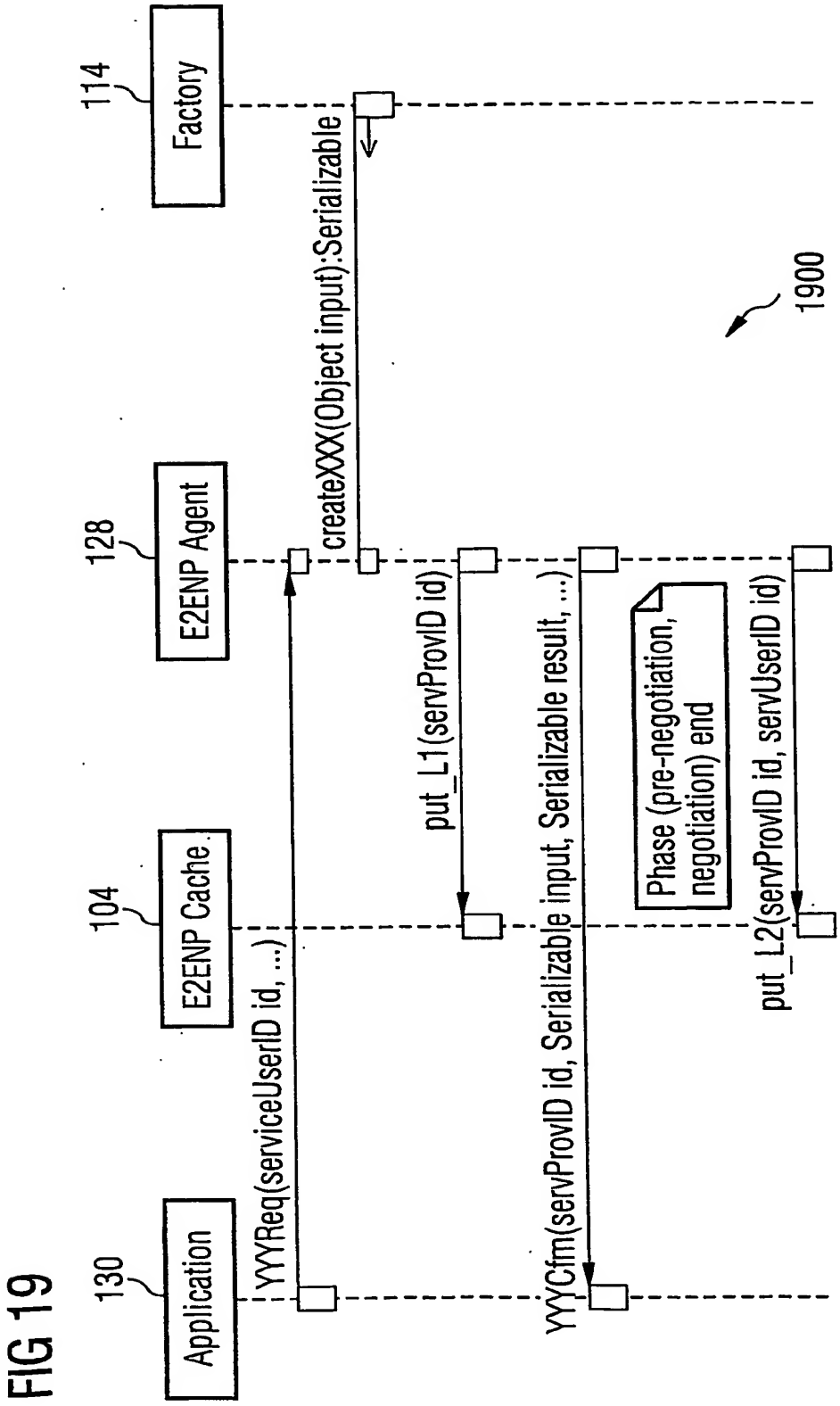

114
Factory
createXXX(Object input):Serializable
1900
128
E2ENP Agent
put_L1(servProvID id)
Phase (pre-negotiation, negotiation) end
104
E2ENP Cache
YYYReq(serviceUserID id, ...)
YYYCfm(servProvID id, Serializable input, Serializable result, ...)
put_L2(servProvID id, servUserID id)
130
Application

FIG 19

FIG 20

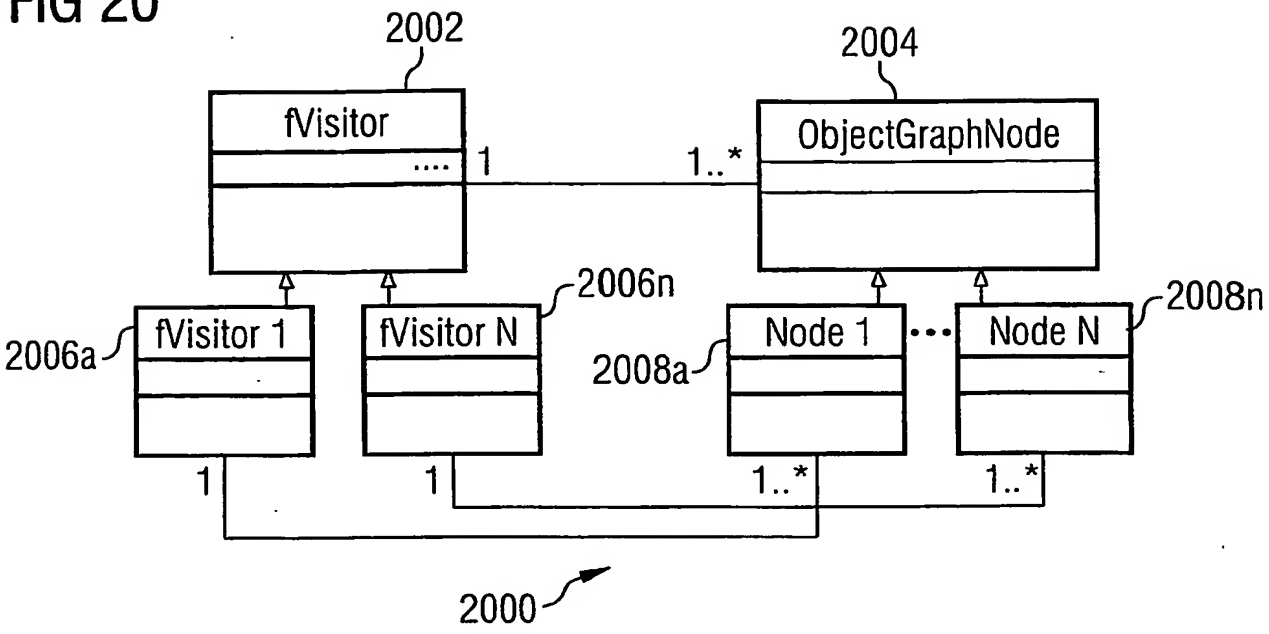

2002
2004
fVisitor
....
1
1..*
ObjectGraphNode
2006n
2008n
2006a
fVisitor 1
fVisitor N
2008a
Node 1
Node N
1
1
1..*
1..*
2000

FIG 21

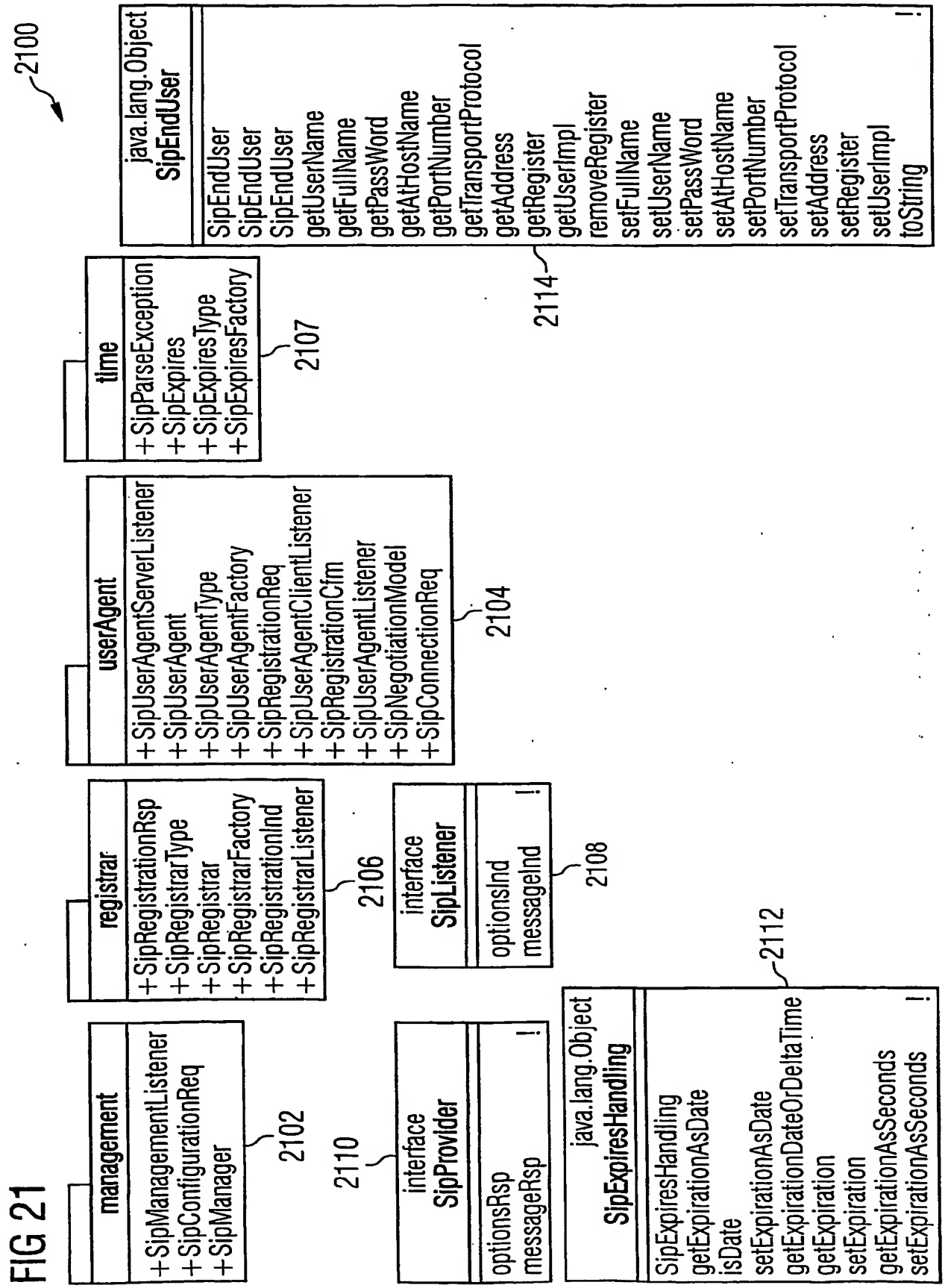

2100
management
+SipManagementListener
+SipConfigurationReq
+SipManager
2102
registrar
+SipRegistrationRsp
+SipRegistrarType
+SipRegistrar
+SipRegistrarFactory
+SipRegistrationInd
+SipRegistrarListener
2106
userAgent
+SipUserAgentServerListener
+SipUserAgent
+SipUserAgentType
+SipUserAgentFactory
+SipRegistrationReq
+SipUserAgentClientListener
+SipRegistrationCfm
+SipUserAgentListener
+SipNegotiationModel
+SipConnectionReq
2104
time
+SipParseException
+SipExpires
+SipExpiresType
+SipExpiresFactory
2107
java.lang.Object
SipEndUser
SipEndUser
SipEndUser
SipEndUser
getUserName
getFullName
getPassWord
getAtHostName
getPortNumber
getTransportProtocol
getAddress
getRegister
getUserImpl
removeRegister
setFullName
setUserName
setPassWord
setAtHostName
setPortNumber
setTransportProtocol
setAddress
setRegister
setUserImpl
toString
!
2114
interface
SipProvider
optionsRsp
messageRsp
!
2110
interface
SipListener
optionsInd
messageInd
!
2108
java.lang.Object
SipExpiresHandling
SipExpiresHandling
getExpirationAsDate
isDate
setExpirationAsDate
getExpirationDateOrDeltaTime
getExpiration
setExpiration
getExpirationAsSeconds
setExpirationAsSeconds
!
2112

FIG 22

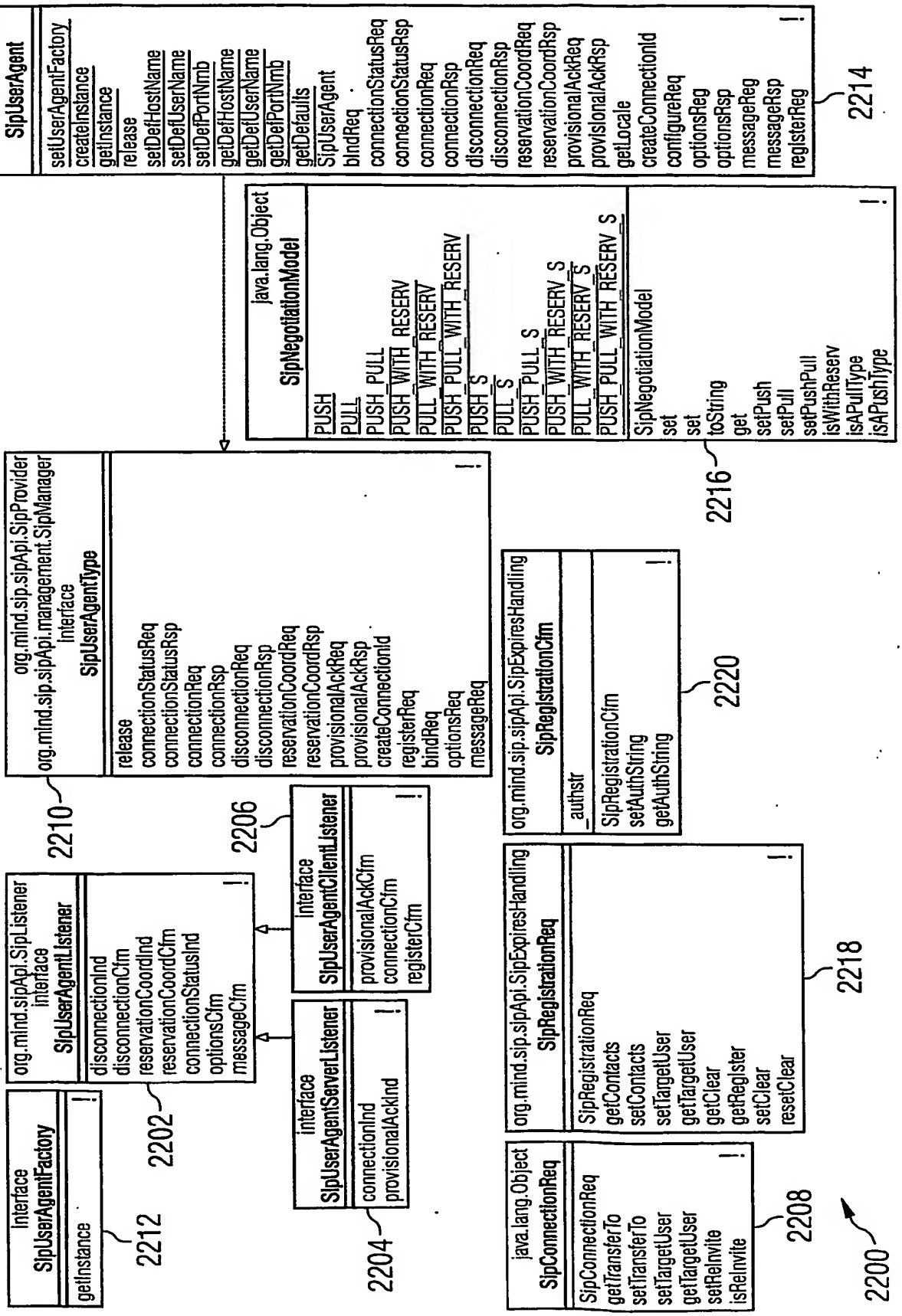

java.lang.Object
SipUserAgent
setUserAgentFactory
createInstance
getInstance
release
setDefHostName
setDefUserName
setDefPortNmb
getDefHostName
getDefUserName
getDefPortNmb
getDefaults
SipUserAgent
bindReq
connectionStatusReq
connectionStatusRsp
connectionReq
connectionRsp
disconnectionReq
disconnectionRsp
reservationCoordReq
reservationCoordRsp
provisionalAckReq
provisionalAckRsp
getLocale
createConnectionId
configureReq
optionsReq
optionsRsp
messageReq
messageRsp
registerReq
2214
java.lang.Object
SipNegotiationModel
PUSH
PULL
PUSH_PULL
PUSH_WITH_RESERV
PULL_WITH_RESERV
PUSH_PULL_WITH_RESERV
PUSH_S
PULL_S
PUSH_PULL_S
PUSH_WITH_RESERV_S
PULL_WITH_RESERV_S
PUSH_PULL_WITH_RESERV_S
SipNegotiationModel
set
set
toString
get
setPush
setPull
setPushPull
isWithReserv
isAPullType
isAPushType
2216
org.mind.sip.sipApi.SipProvider
org.mind.sip.sipApi.management.SipManager
Interface
SipUserAgentType
release
connectionStatusReq
connectionStatusRsp
connectionReq
connectionRsp
disconnectionReq
disconnectionRsp
reservationCoordReq
reservationCoordRsp
provisionalAckReq
provisionalAckRsp
createConnectionId
registerReq
bindReq
optionsReq
messageReq
2210
org.mind.sip.sipApi.SipListener
interface
SipUserAgentListener
disconnectionInd
disconnectionCfm
reservationCoordInd
reservationCoordCfm
connectionStatusInd
optionsCfm
messageCfm
2202
Interface
SipUserAgentFactory
getInstance
2212
interface
SipUserAgentServerListener
connectionInd
provisionalAckInd
2204
Interface
SipUserAgentClientListener
provisionalAckCfm
connectionCfm
registerCfm
2206
org.mind.sip.sipApi.SipExpiresHandling
SipRegistrationCfm
_authstr
SipRegistrationCfm
setAuthString
getAuthString
2220
org.mind.sip.sipApi.SipExpiresHandling
SipRegistrationReq
SipRegistrationReq
getContacts
setContacts
setTargetUser
getTargetUser
getClear
getRegister
setClear
resetClear
2218
java.lang.Object
SipConnectionReq
SipConnectionReq
getTransferTo
setTransferTo
setTargetUser
getTargetUser
setReInvite
isReInvite
2208
2200

FIG 23

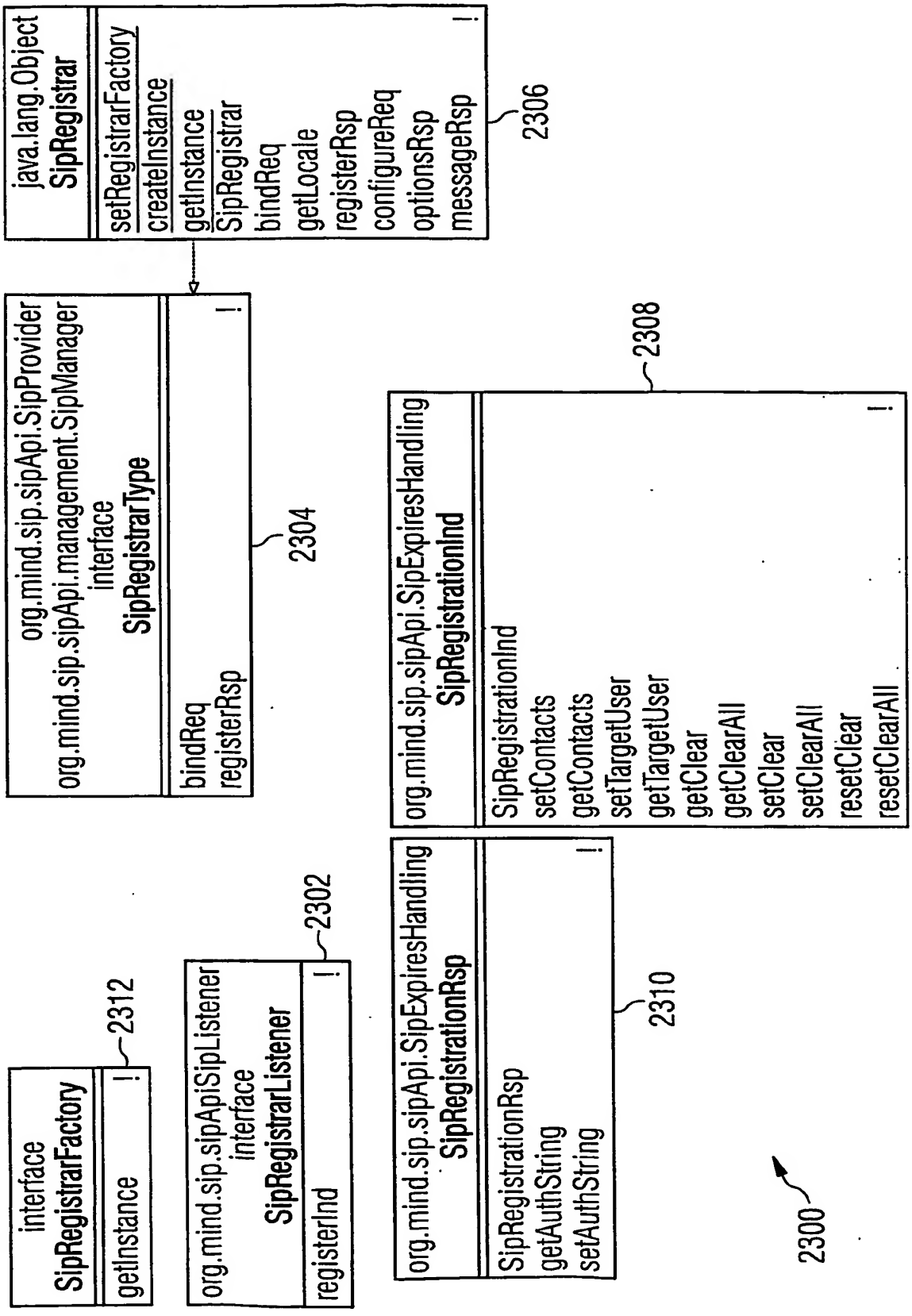

java.lang.Object
SipRegistrar
setRegistrarFactory
createInstance
getInstance
SipRegistrar
bindReq
getLocale
registerRsp
configureReq
optionsRsp
messageRsp
!
2306
org.mind.sip.sipApi.SipProvider
org.mind.sip.sipApi.management.SipManager
interface
SipRegistrarType
bindReq
registerRsp
!
2304
org.mind.sip.sipApi.SipExpiresHandling
SipRegistrationInd
SipRegistrationInd
setContacts
getContacts
setTargetUser
getTargetUser
getClear
getClearAll
setClear
setClearAll
resetClear
resetClearAll
!
2308
interface
SipRegistrarFactory
getInstance
!
2312
org.mind.sip.sipApiSipListener
interface
SipRegistrarListener
registerInd
!
2302
org.mind.sip.sipApi.SipExpiresHandling
SipRegistrationRsp
SipRegistrationRsp
getAuthString
setAuthString
!
2310
2300

FIG 24

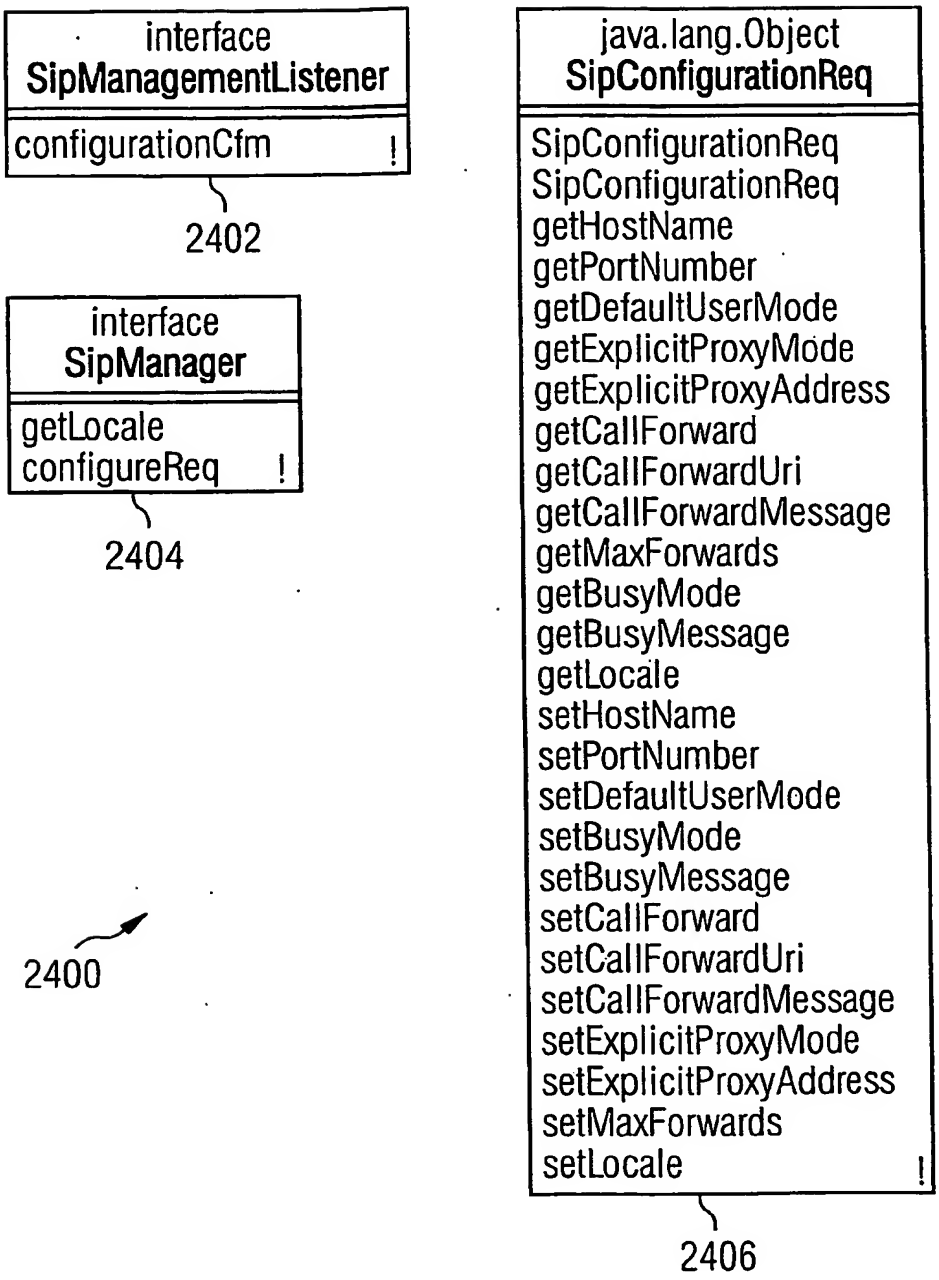
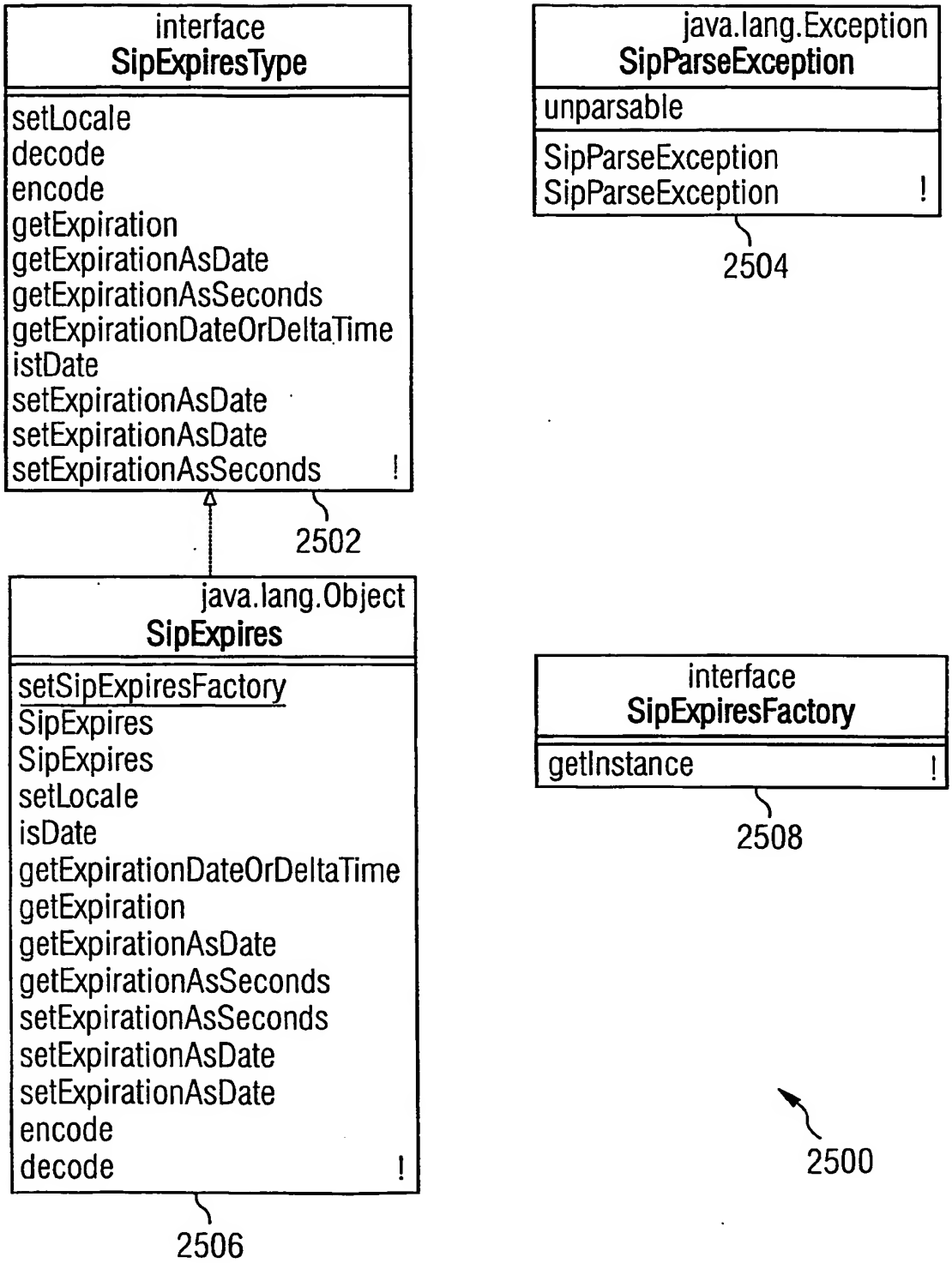

interface
SipManagementListener
configurationCfm !
2402
interface
SipManager
getLocale
configureReq !
2404
2400
java.lang.Object
SipConfigurationReq
SipConfigurationReq
SipConfigurationReq
getHostName
getPortNumber
getDefaultUserMode
getExplicitProxyMode
getExplicitProxyAddress
getCallForward
getCallForwardUri
getCallForwardMessage
getMaxForwards
getBusyMode
getBusyMessage
getLocale
setHostName
setPortNumber
setDefaultUserMode
setBusyMode
setBusyMessage
setCallForward
setCallForwardUri
setCallForwardMessage
setExplicitProxyMode
setExplicitProxyAddress
setMaxForwards
setLocale !
2406

FIG 25


interface
SipExpiresType
setLocale
decode
encode
getExpiration
getExpirationAsDate
getExpirationAsSeconds
getExpirationDateOrDeltaTime
istDate
setExpirationAsDate
setExpirationAsDate
setExpirationAsSeconds !
2502
java.lang.Exception
SipParseException
unparsable
SipParseException
SipParseException !
2504
java.lang.Object
SipExpires
setSipExpiresFactory
SipExpires
SipExpires
setLocale
isDate
getExpirationDateOrDeltaTime
getExpiration
getExpirationAsDate
getExpirationAsSeconds
setExpirationAsSeconds
setExpirationAsDate
setExpirationAsDate
encode
decode !
2506
interface
SipExpiresFactory
getInstance !
2508
2500

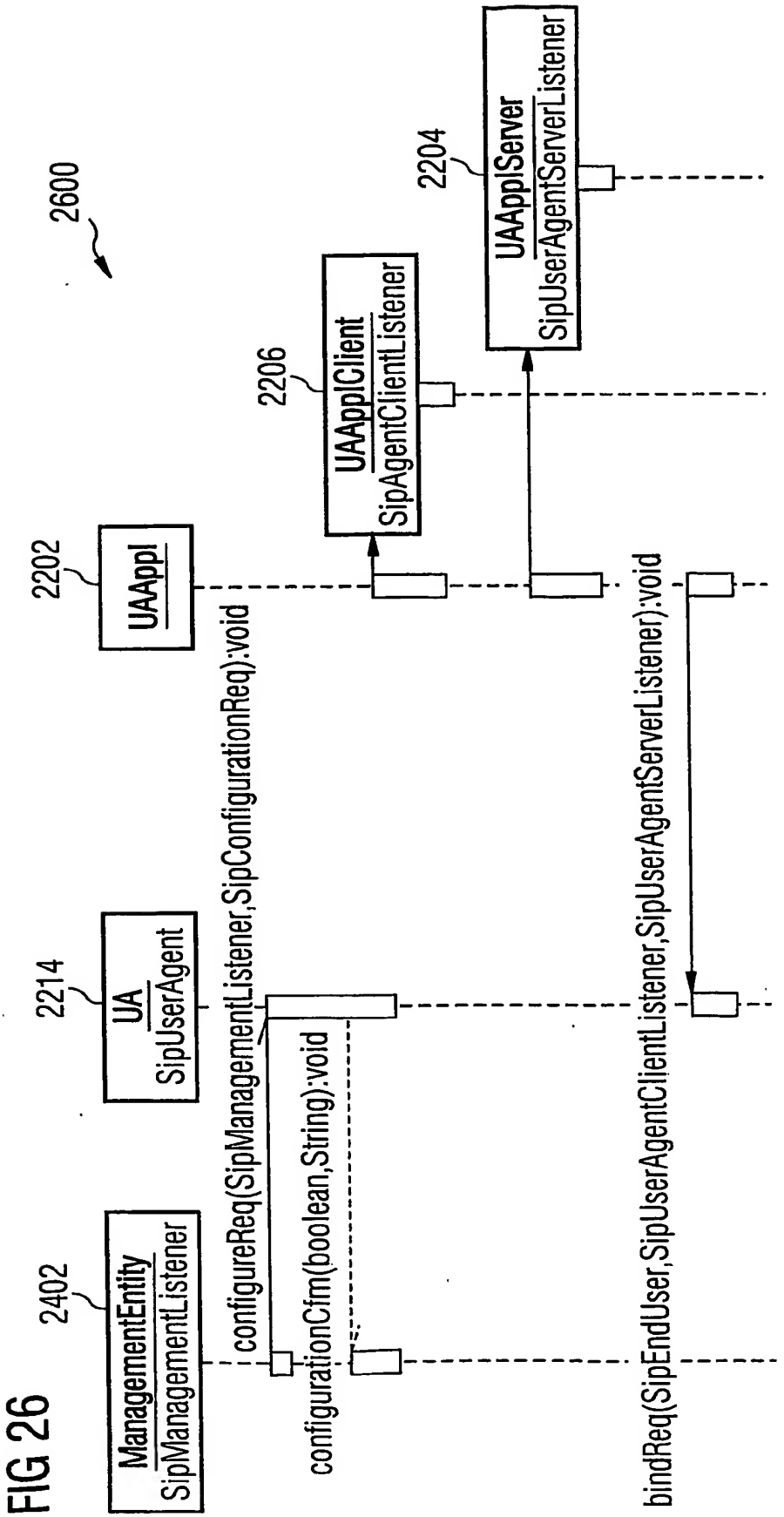
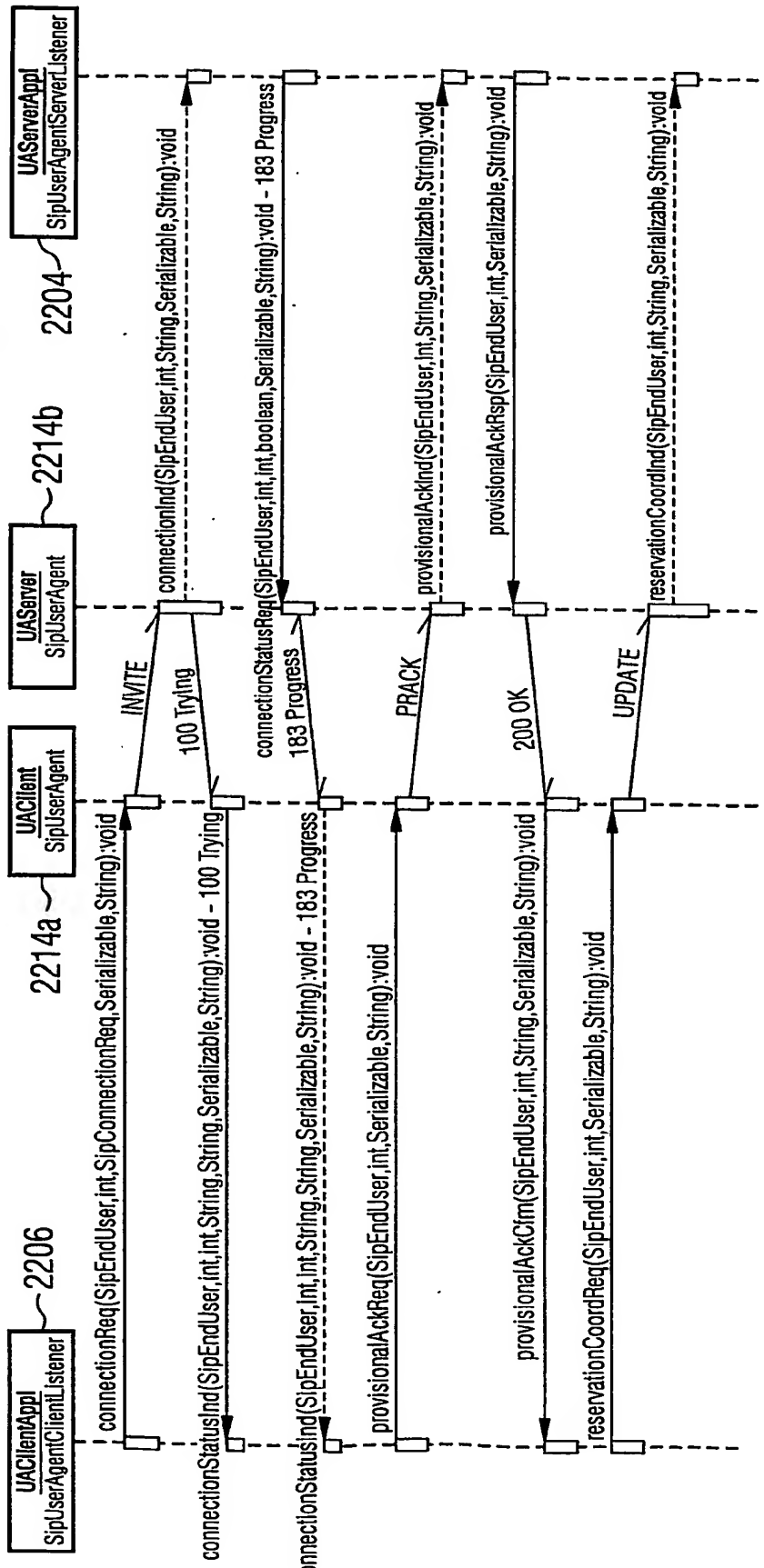

2600
2402
ManagementEntity
SipManagementListener
2214
UA
SipUserAgent
2202
UAAppl
2206
UAApplClient
SipAgentClientListener
2204
UAApplServer
SipUserAgentServerListener
configureReq(SipManagementListener,SipConfigurationReq):void
configurationCfm(boolean,String):void
bindReq(SipEndUser,SipUserAgentClientListener,SipUserAgentServerListener):void

FIG 26

FIG 27
(Part 1)


2700
UAClientAppl
SipUserAgentClientListener
2206
UAClient
SipUserAgent
2214a
UAServer
SipUserAgent
2214b
UAServerAppl
SipUserAgentServerListener
2204
connectionReq(SipEndUser,int,SipConnectionReq,Serializable,String):void
INVITE
connectionInd(SipEndUser,int,String,Serializable,String):void
100 Trying
connectionStatusInd(SipEndUser,int,int,String,Serializable,String):void - 100 Trying
connectionStatusReq(SipEndUser,int,int,boolean,Serializable,String):void - 183 Progress
183 Progress
connectionStatusInd(SipEndUser,int,int,String,Serializable,String):void - 183 Progress
provisionalAckReq(SipEndUser,int,Serializable,String):void
PRACK
provisionalAckInd(SipEndUser,int,String,Serializable,String):void
provisionalAckRsp(SipEndUser,int,Serializable,String):void
200 OK
provisionalAckCfm(SipEndUser,int,String,Serializable,String):void
reservationCoordReq(SipEndUser,int,Serializable,String):void
UPDATE
reservationCoordInd(SipEndUser,int,String,Serializable,String):void

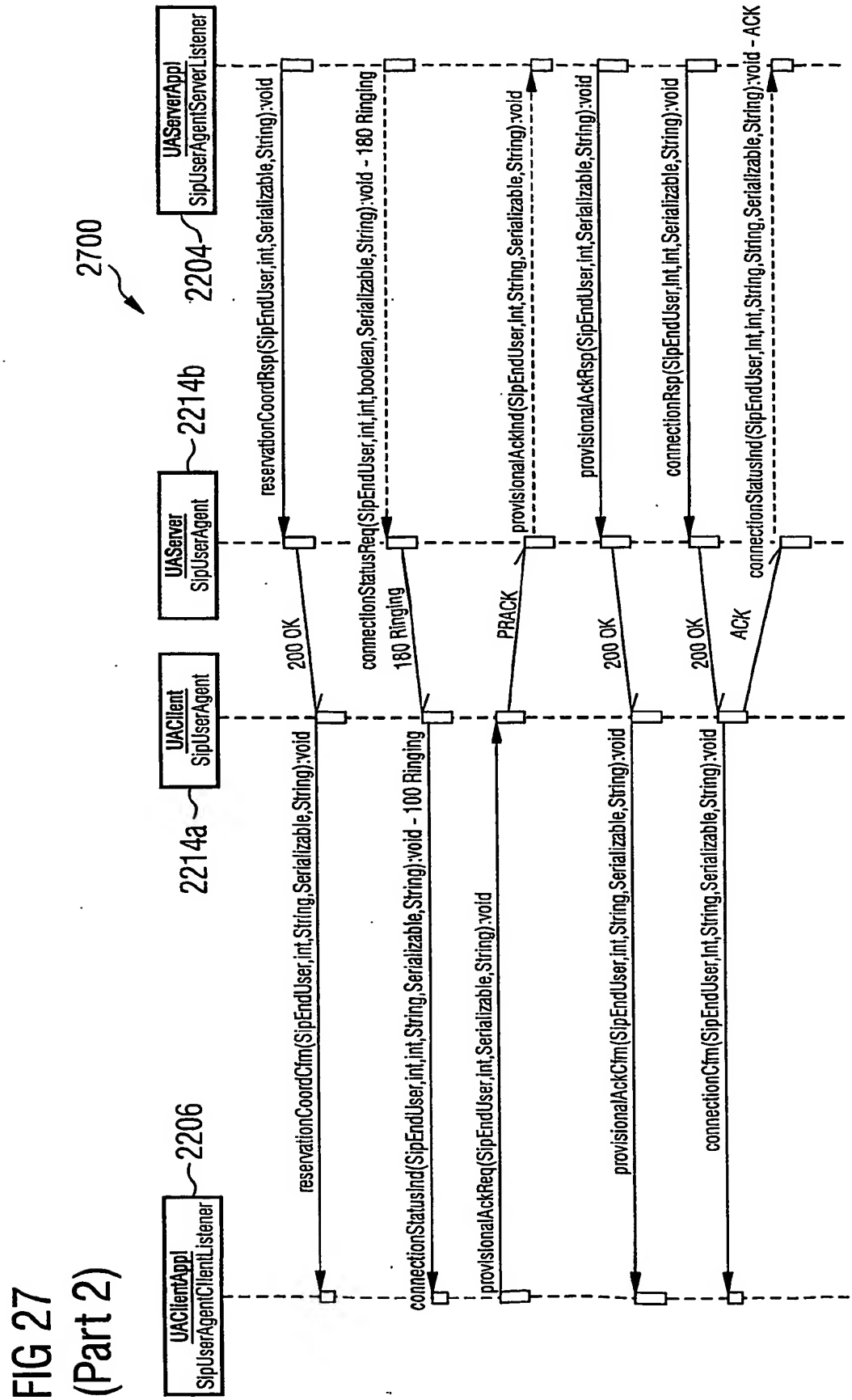

2700
UAClientAppl
SipUserAgentClientListener
2206
UAClient
SipUserAgent
2214a
UAServer
SipUserAgent
2214b
UAServerAppl
SipUserAgentServerListener
2204
reservationCoordCfm(SipEndUser,int,String,Serializable,String):void
reservationCoordRsp(SipEndUser,int,Serializable,String):void
200 OK
connectionStatusInd(SipEndUser,int,int,String,Serializable,String):void - 100 Ringing
connectionStatusReq(SipEndUser,int,int,boolean,Serializable,String):void - 180 Ringing
180 Ringing
provisionalAckReq(SipEndUser,int,Serializable,String):void
provisionalAckInd(SipEndUser,int,String,Serializable,String):void
PRACK
provisionalAckCfm(SipEndUser,int,String,Serializable,String):void
provisionalAckRsp(SipEndUser,int,Serializable,String):void
200 OK
connectionCfm(SipEndUser,int,String,Serializable,String):void
connectionRsp(SipEndUser,int,int,Serializable,String):void
200 OK
connectionStatusInd(SipEndUser,int,int,String,String,Serializable,String):void - ACK
ACK

FIG 27
(Part 2)

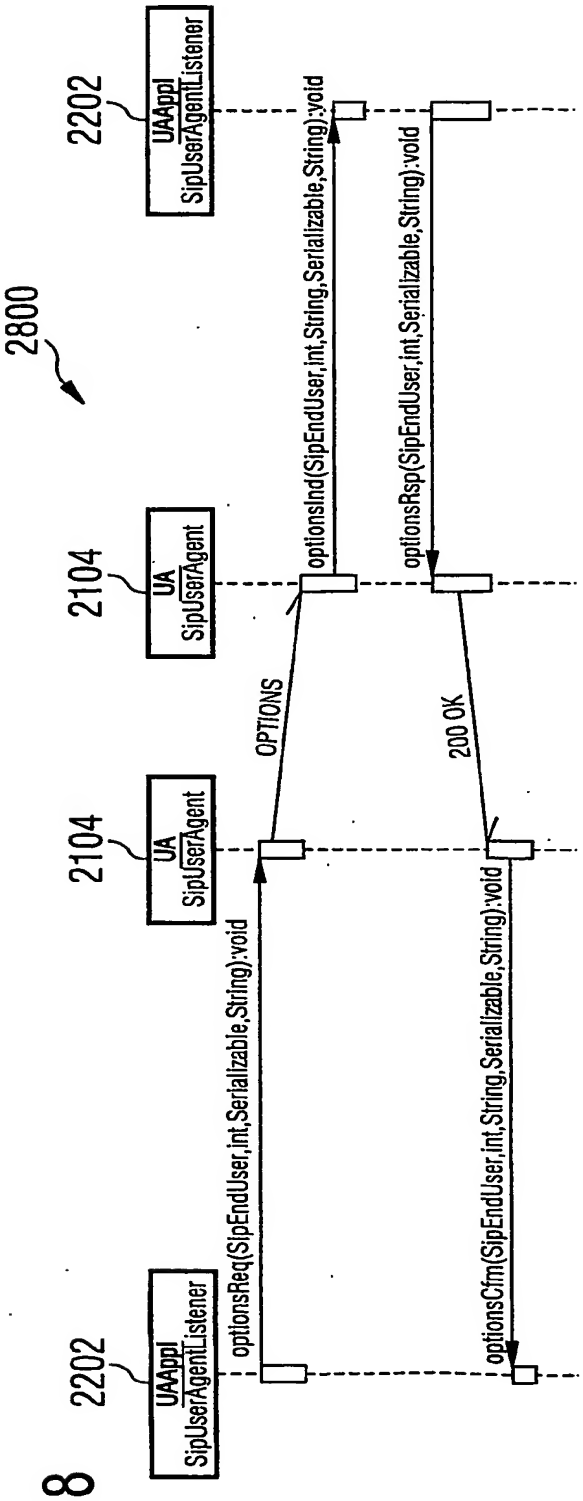

2800
2202
UAAppl
SipUserAgentListener
2104
UA
SipUserAgent
2104
UA
SipUserAgent
2202
UAAppl
SipUserAgentListener
optionsReq(SipEndUser,int,Serializable,String):void
OPTIONS
optionsInd(SipEndUser,int,String,Serializable,String):void
200 OK
optionsRsp(SipEndUser,int,Serializable,String):void
optionsCfm(SipEndUser,int,String,Serializable,String):void

FIG 28

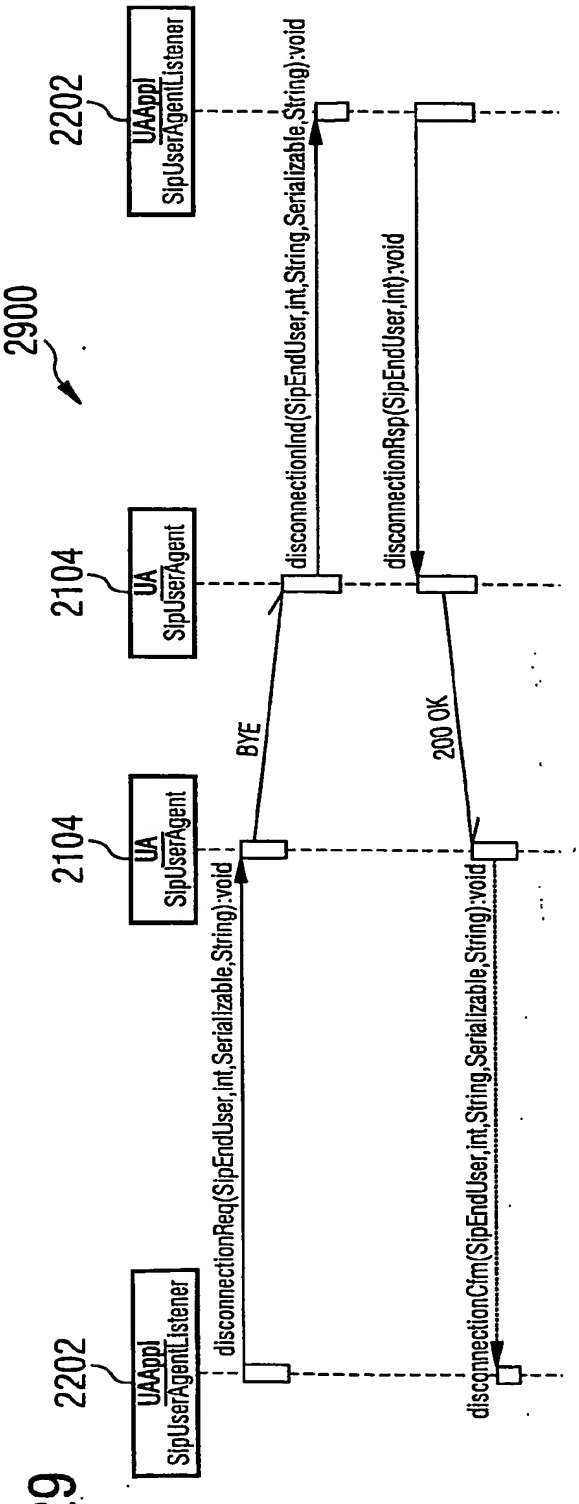

2900
2202
UAAppl
SipUserAgentListener
2104
UA
SipUserAgent
2104
UA
SipUserAgent
2202
UAAppl
SipUserAgentListener
disconnectionReq(SipEndUser,int,Serializable,String):void
BYE
disconnectionInd(SipEndUser,int,String,Serializable,String):void
200 OK
disconnectionRsp(SipEndUser,int):void
disconnectionCfm(SipEndUser,int,String,Serializable,String):void

FIG 29

FIG 30

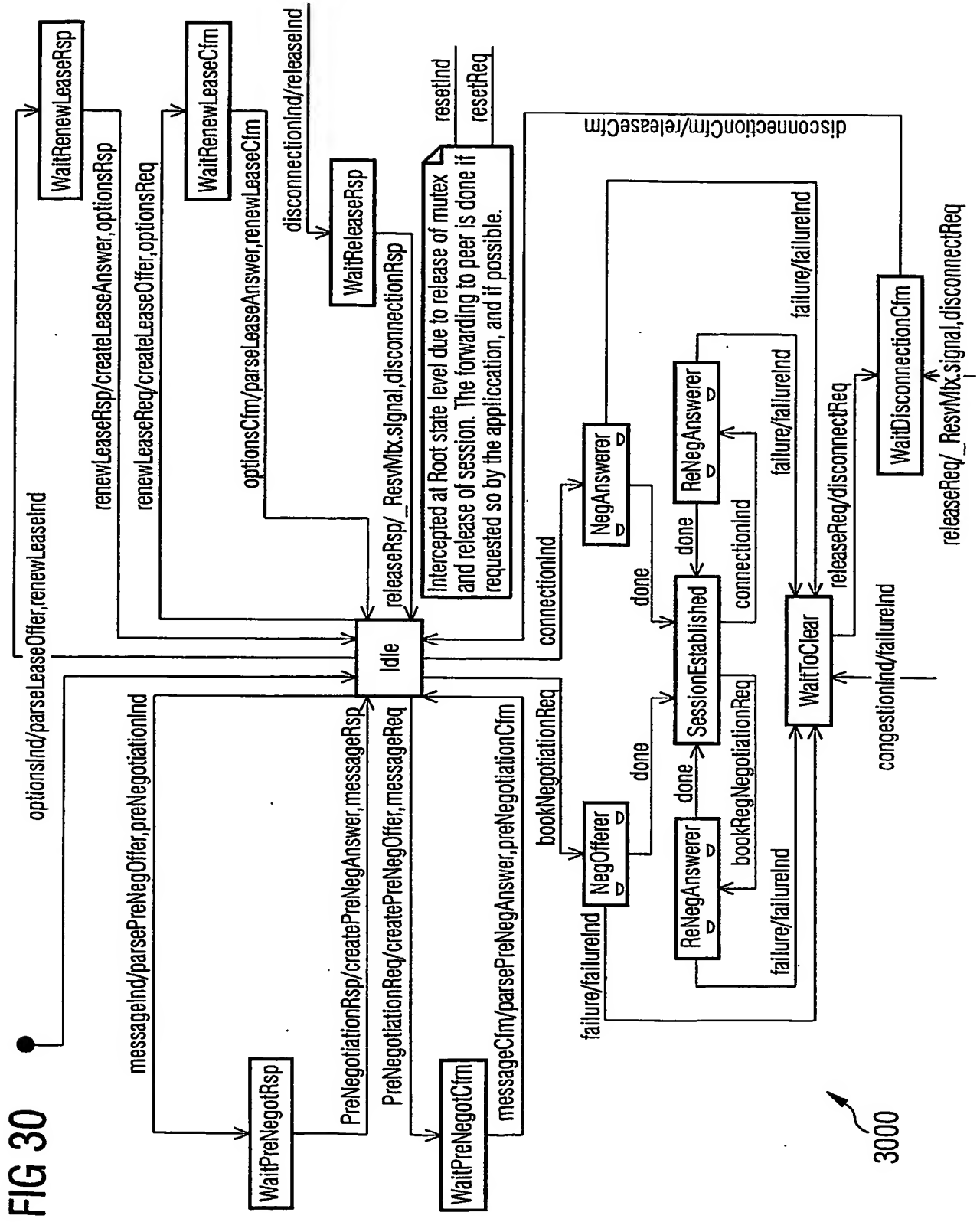

optionsInd/parseLeaseOffer,renewLeaseInd
WaitRenewLeaseRsp
renewLeaseRsp/createLeaseAnswer,optionsRsp
renewLeaseReq/createLeaseOffer,optionsReq
WaitRenewLeaseCfm
optionsCfm/parseLeaseAnswer,renewLeaseCfm
disconnectInd/releaseInd
WaitReleaseRsp
releaseRsp/_ResvMtx.signal,disconnectRsp
resetInd
resetReq
Intercepted at Root state level due to release of mutex and release of session. The forwarding to peer is done if requested so by the applicaation, and if possible.
disconnectionCfm/releaseCfm
Idle
connectionInd
NegAnswerer
done
ReNegAnswerer
done
connectionInd
failure/failureInd
failure/failureInd
messageInd/parsePreNegOffer,preNegotiationInd
PreNegotiationRsp/createPreNegAnswer,messageRsp
WaitPreNegotRsp
PreNegotiationReq/createPreNegOffer,messageReq
WaitPreNegotCfm
messageCfm/parsePreNegAnswer,preNegotiationCfm
bookNegotiationReq
NegOfferer
done
SessionEstablished
done
bookRegNegotiationReq
ReNegAnswerer
failure/failureInd
failure/failureInd
WaitToClear
releaseReq/disconnectReq
WaitDisconnectionCfm
releaseReq/_ResvMtx.signal,disconnectReq
congestionInd/failureInd
3000

FIG 31

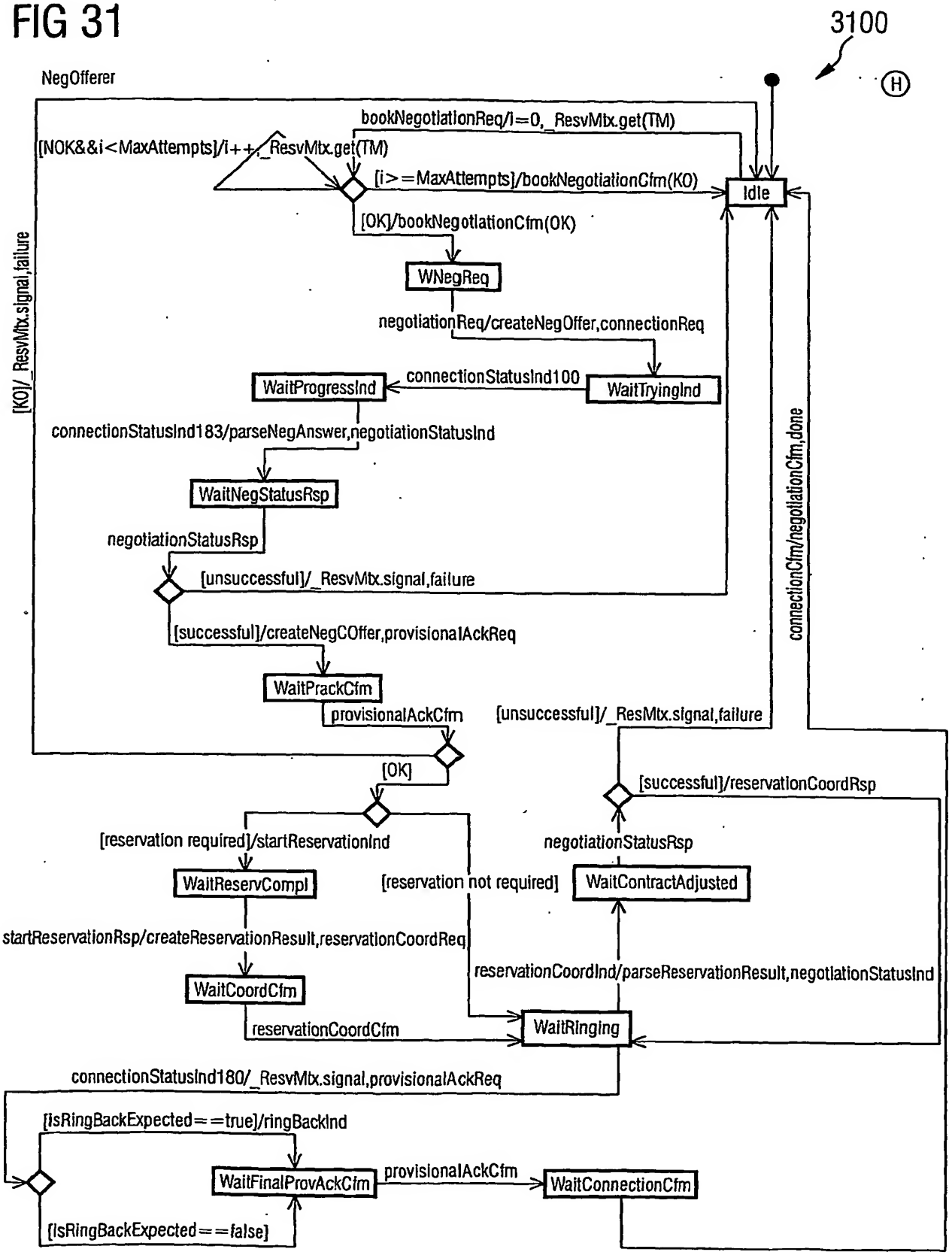

3100
NegOfferer
H
bookNegotiationReq/i=0,_ResvMtx.get(TM)
[NOK&&i<MaxAttempts]/i++,_ResvMtx.get(TM)
[i>=MaxAttempts]/bookNegotiationCfm(KO)
Idle
[OK]/bookNegotiationCfm(OK)
WNegReq
negotiationReq/createNegOffer,connectionReq
WaitProgressInd
connectionStatusInd100
WaitTryingInd
connectionStatusInd183/parseNegAnswer,negotiationStatusInd
WaitNegStatusRsp
negotiationStatusRsp
[unsuccessful]/_ResvMtx.signal,failure
[successful]/createNegCOffer,provisionalAckReq
WaitPrackCfm
provisionalAckCfm
[unsuccessful]/_ResMtx.signal,failure
[KO]/_ResvMtx.signal,failure
connectionCfm/negotiationCfm,done
[OK]
[successful]/reservationCoordRsp
[reservation required]/startReservationInd
negotiationStatusRsp
WaitReservCompl
[reservation not required]
WaitContractAdjusted
startReservationRsp/createReservationResult,reservationCoordReq
reservationCoordInd/parseReservationResult,negotiationStatusInd
WaitCoordCfm
reservationCoordCfm
WaitRinging
connectionStatusInd180/_ResvMtx.signal,provisionalAckReq
[isRingBackExpected==true]/ringBackInd
WaitFinalProvAckCfm
provisionalAckCfm
WaitConnectionCfm
[isRingBackExpected==false]

FIG 32

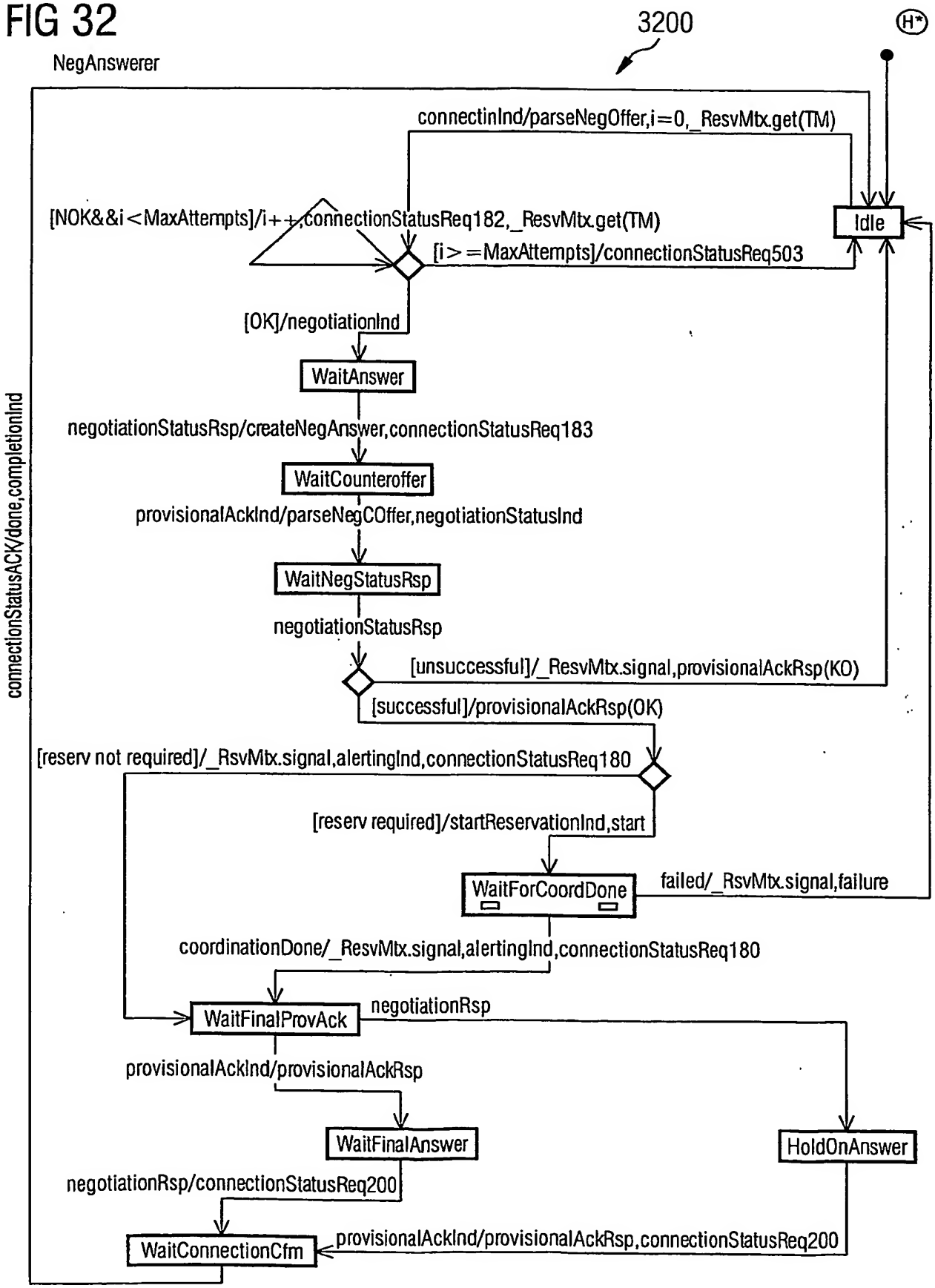

3200
H*
NegAnswerer
connectionStatusACK/done,completionInd
connectinInd/parseNegOffer,i=0,_ResvMtx.get(TM)
[NOK&&i<MaxAttempts]/i++,connectionStatusReq182,_ResvMtx.get(TM)
[i>=MaxAttempts]/connectionStatusReq503
Idle
[OK]/negotiationInd
WaitAnswer
negotiationStatusRsp/createNegAnswer,connectionStatusReq183
WaitCounteroffer
provisionalAckInd/parseNegCOffer,negotiationStatusInd
WaitNegStatusRsp
negotiationStatusRsp
[unsuccessful]/_ResvMtx.signal,provisionalAckRsp(KO)
[successful]/provisionalAckRsp(OK)
[reserv not required]/_RsvMtx.signal,alertingInd,connectionStatusReq180
[reserv required]/startReservationInd,start
WaitForCoordDone
failed/_RsvMtx.signal,failure
coordinationDone/_ResvMtx.signal,alertingInd,connectionStatusReq180
WaitFinalProvAck
negotiationRsp
provisionalAckInd/provisionalAckRsp
WaitFinalAnswer
HoldOnAnswer
negotiationRsp/connectionStatusReq200
WaitConnectionCfm
provisionalAckInd/provisionalAckRsp,connectionStatusReq200

FIG 33

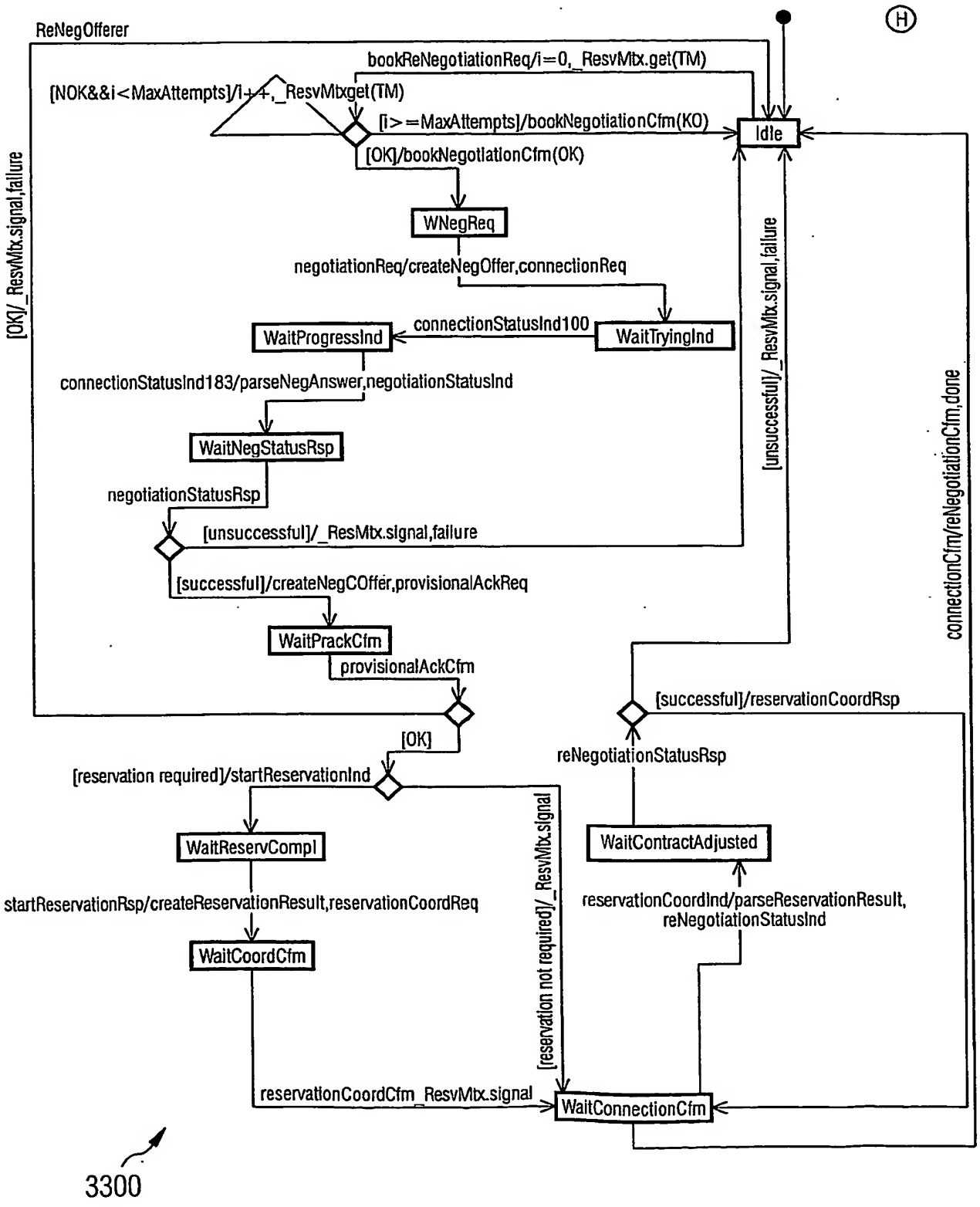

ReNegOfferer
H
bookReNegotiationReq/i=0,_ResvMtx.get(TM)
[NOK&&i<MaxAttempts]/i++,_ResvMtxget(TM)
[i>=MaxAttempts]/bookNegotiationCfm(KO)
Idle
[OK]/bookNegotiationCfm(OK)
WNegReq
negotiationReq/createNegOffer,connectionReq
WaitProgressInd
connectionStatusInd100
WaitTryingInd
connectionStatusInd183/parseNegAnswer,negotiationStatusInd
WaitNegStatusRsp
negotiationStatusRsp
[unsuccessful]/_ResMtx.signal,failure
[successful]/createNegCOffer,provisionalAckReq
WaitPrackCfm
provisionalAckCfm
[OK]/_ResvMtx.signal,failure
[unsuccessful]/_ResvMtx.signal,failure
connectionCfm/reNegotiationCfm,done
[successful]/reservationCoordRsp
[OK]
reNegotiationStatusRsp
[reservation required]/startReservationInd
WaitReservCompl
WaitContractAdjusted
startReservationRsp/createReservationResult,reservationCoordReq
reservationCoordInd/parseReservationResult,
reNegotiationStatusInd
WaitCoordCfm
[reservation not required]/_ResvMtx.signal
reservationCoordCfm_ResvMtx.signal
WaitConnectionCfm
3300

FIG 34

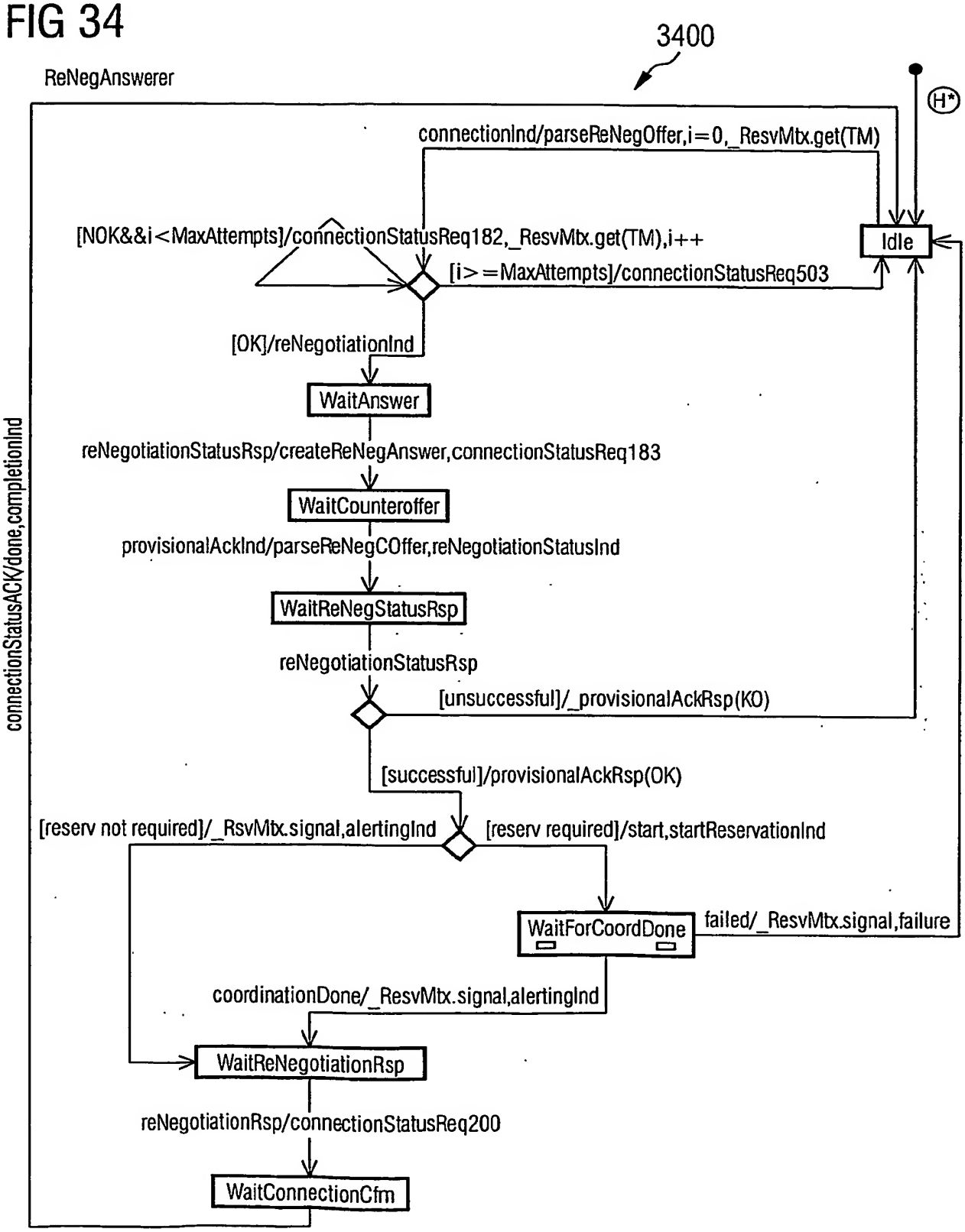

3400
ReNegAnswerer
H*
connectionInd/parseReNegOffer,i=0,_ResvMtx.get(TM)
Idle
[NOK&&i<MaxAttempts]/connectionStatusReq182,_ResvMtx.get(TM),i++
[i>=MaxAttempts]/connectionStatusReq503
[OK]/reNegotiationInd
WaitAnswer
reNegotiationStatusRsp/createReNegAnswer,connectionStatusReq183
WaitCounteroffer
provisionalAckInd/parseReNegCOffer,reNegotiationStatusInd
WaitReNegStatusRsp
reNegotiationStatusRsp
[unsuccessful]/_provisionalAckRsp(KO)
[successful]/provisionalAckRsp(OK)
[reserv not required]/_RsvMtx.signal,alertingInd
[reserv required]/start,startReservationInd
WaitForCoordDone
failed/_ResvMtx.signal,failure
coordinationDone/_ResvMtx.signal,alertingInd
WaitReNegotiationRsp
reNegotiationRsp/connectionStatusReq200
WaitConnectionCfm
connectionStatusACK/done,completionInd

FIG 35

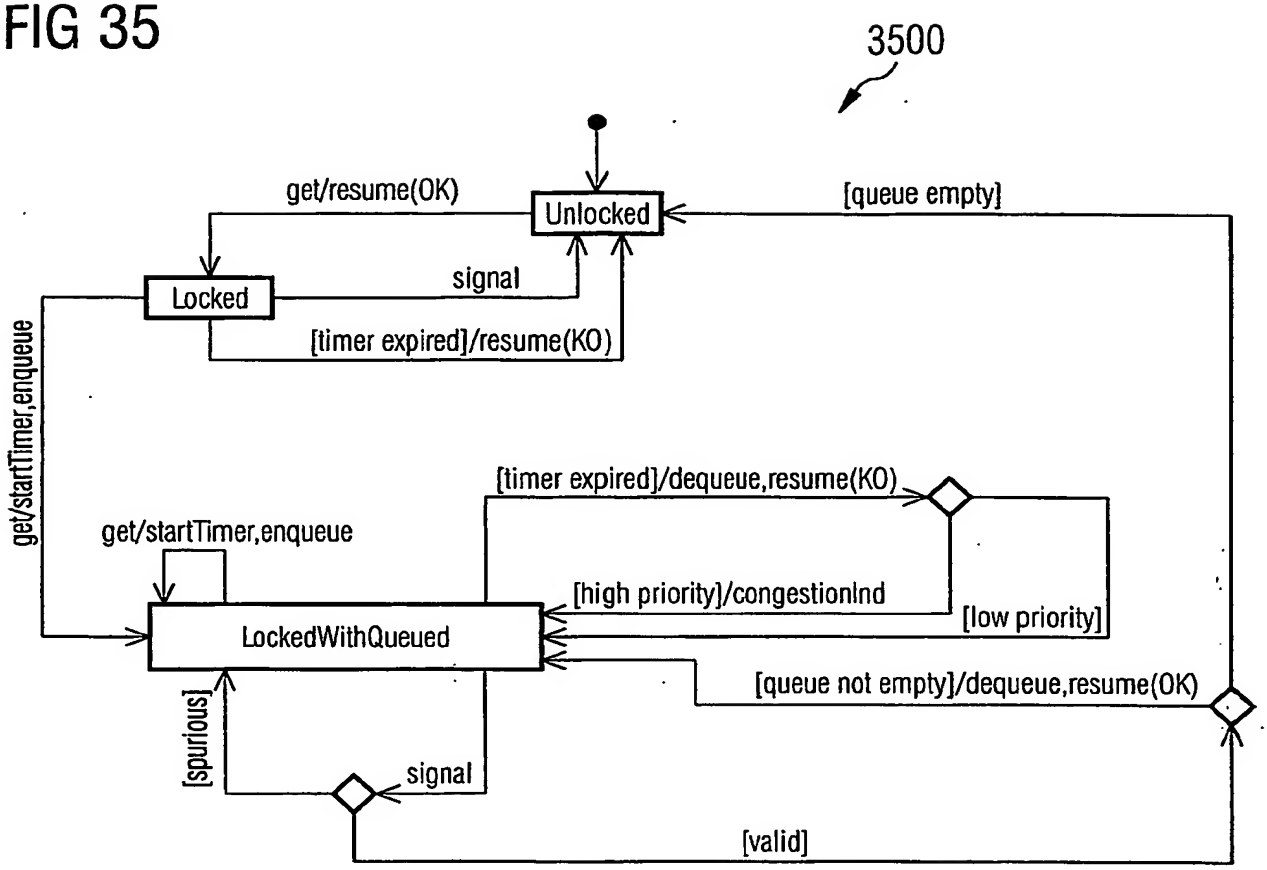
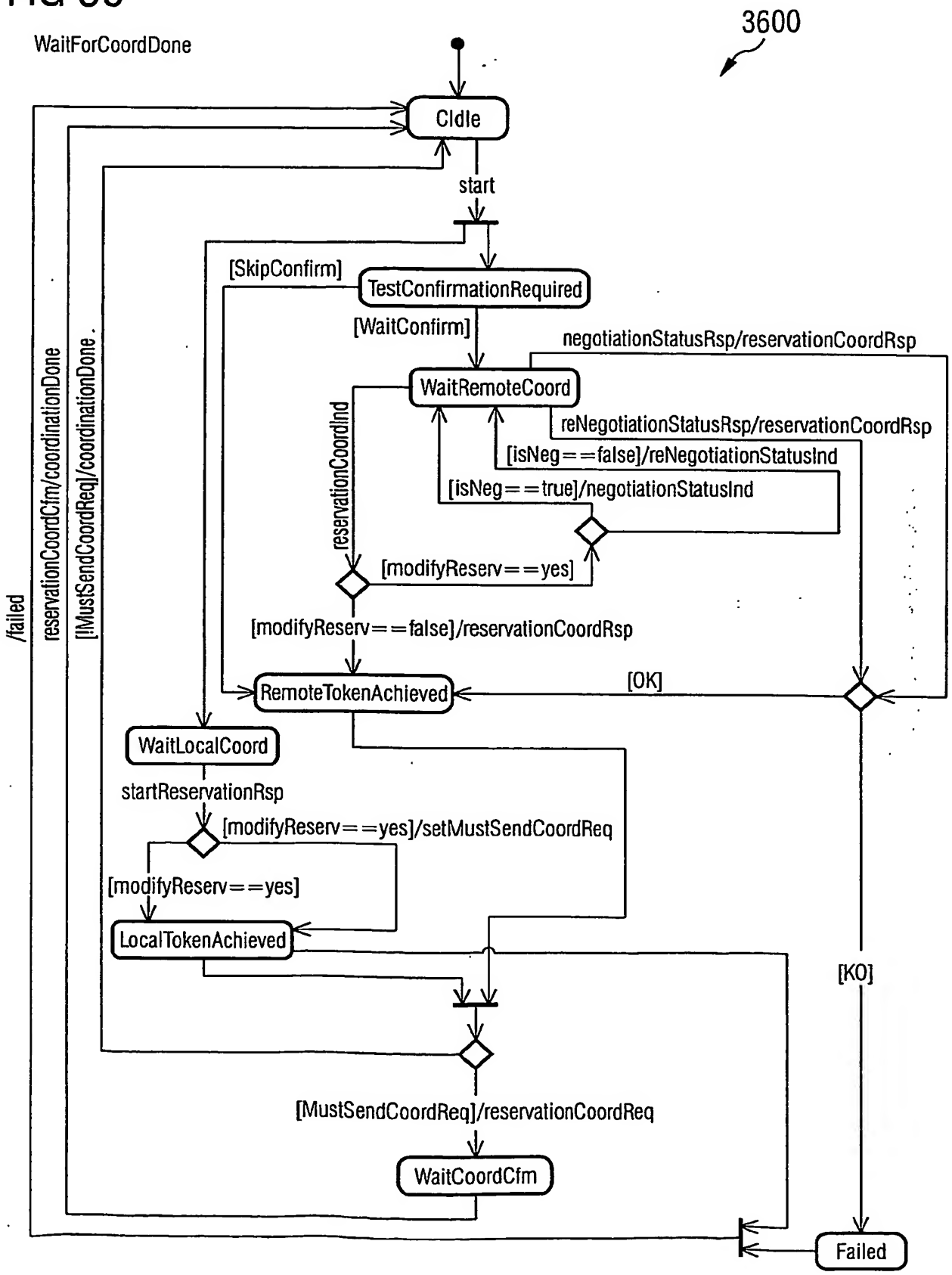

3500
get/resume(OK)
Unlocked
[queue empty]
Locked
signal
[timer expired]/resume(KO)
get/startTimer,enqueue
[timer expired]/dequeue,resume(KO)
get/startTimer,enqueue
[high priority]/congestionInd
[low priority]
LockedWithQueued
[queue not empty]/dequeue,resume(OK)
[spurious]
signal
[valid]

FIG 36


3600
WaitForCoordDone
CIdle
start
[SkipConfirm]
TestConfirmationRequired
[WaitConfirm]
negotiationStatusRsp/reservationCoordRsp
WaitRemoteCoord
reNegotiationStatusRsp/reservationCoordRsp
[isNeg==false]/reNegotiationStatusInd
[isNeg==true]/negotiationStatusInd
reservationCoordInd
[modifyReserv==yes]
[modifyReserv==false]/reservationCoordRsp
RemoteTokenAchieved
[OK]
WaitLocalCoord
startReservationRsp
[modifyReserv==yes]/setMustSendCoordReq
[modifyReserv==yes]
LocalTokenAchieved
[KO]
[MustSendCoordReq]/reservationCoordReq
WaitCoordCfm
Failed
/failed
reservationCoordCfm/coordinationDone
[!MustSendCoordReq]/coordinationDone

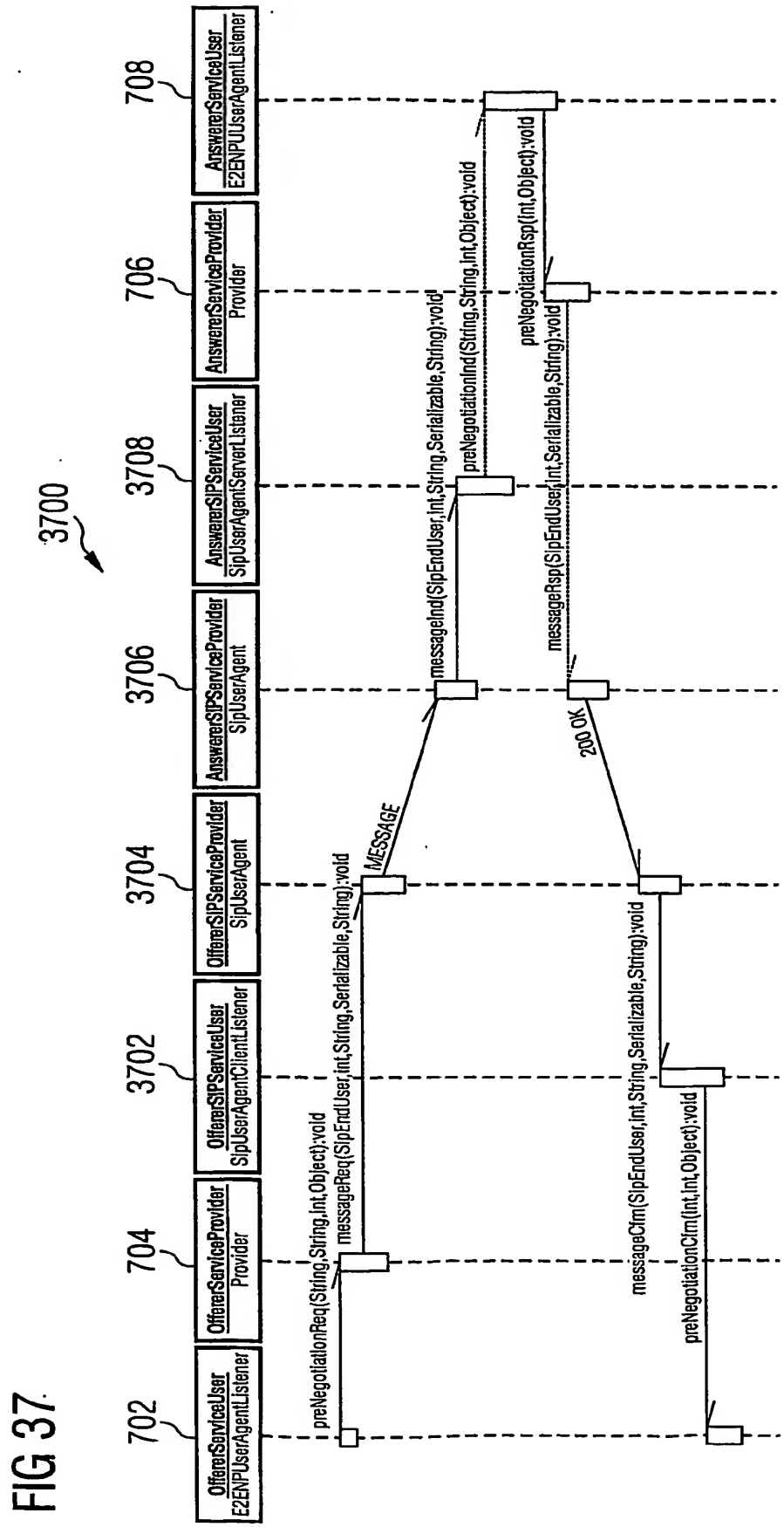

3700
702
OffererUserAgentListener
704
OffererServiceProvider
3702
OffererSIPServiceUser
3704
OffererSIPServiceProvider
3706
AnswererSIPServiceProvider
3708
AnswererSIPServiceUser
706
AnswererServiceProvider
708
AnswererServiceUser
preNegotiationReq(String,String,Int,Object):void
messageReq(SipEndUser,int,String,Serializable,String):void
MESSAGE
messageInd(SipEndUser,Int,String,Serializable,String):void
preNegotiationInd(String,String,Int,Object):void
preNegotiationRsp(Int,Object):void
messageRsp(SipEndUser,Int,Serializable,String):void
200 OK
messageCfm(SipEndUser,int,String,Serializable,String):void
preNegotiationCfm(Int,Int,Object):void

FIG 37

FIG 38
(Part 1)

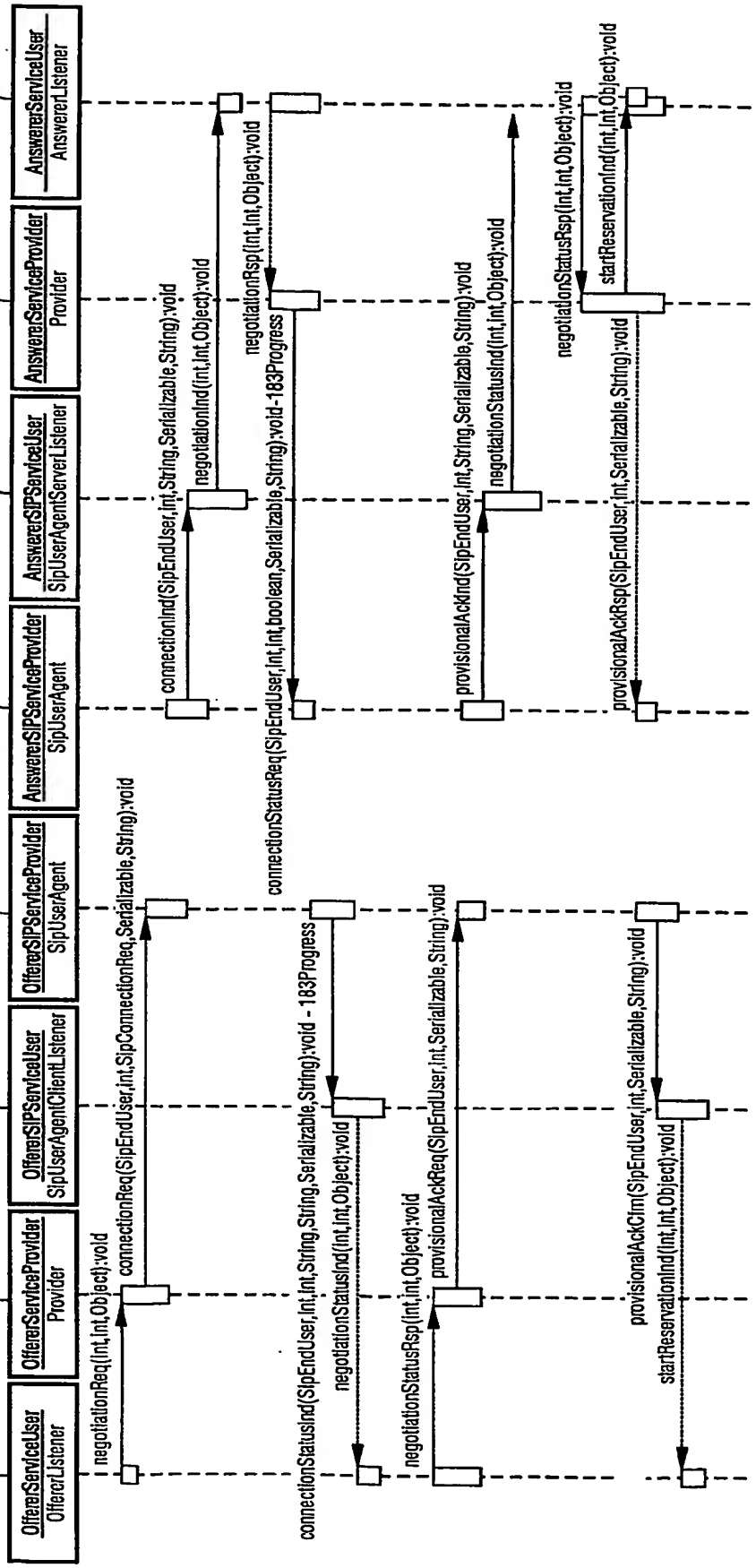
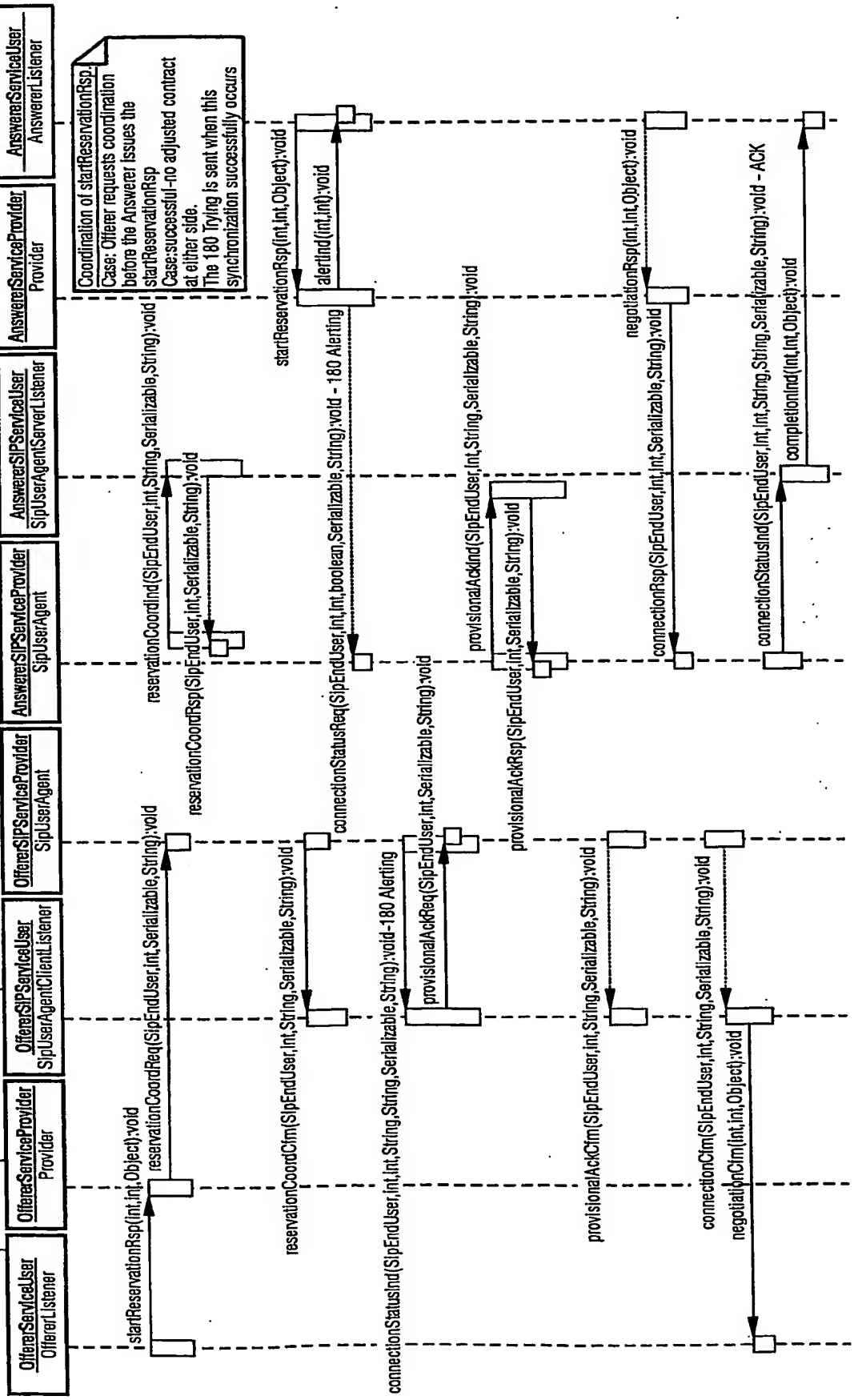

3800
702
704
3702
3704
3706
3708
706
708
OffererServiceUser
OffererListener
OffererServiceProvider
Provider
OffererSIPServiceUser
SipUserAgentClientListener
OffererSIPServiceProvider
SipUserAgent
AnswererSIPServiceProvider
SipUserAgent
AnswererSIPServiceUser
SipUserAgentServerListener
AnswererServiceProvider
Provider
AnswererServiceUser
AnswererListener
negotiationReq(int,int,Object):void
connectionReq(SipEndUser,int,SipConnectionReq,Serializable,String):void
connectionInd(SipEndUser,int,String,Serializable,String):void
negotiationInd(int,int,Object):void
negotiationRsp(int,int,Object):void
connectionStatusReq(SipEndUser,int,int,boolean,Serializable,String):void-183Progress
connectionStatusInd(SipEndUser,int,int,String,String,Serializable,String):void - 183Progress
negotiationStatusInd(int,int,Object):void
provisionalAckReq(SipEndUser,int,Serializable,String):void
negotiationStatusRsp(int,int,Object):void
provisionalAckInd(SipEndUser,int,String,Serializable,String):void
negotiationStatusInd(int,int,Object):void
provisionalAckRsp(SipEndUser,int,Serializable,String):void
negotiationStatusRsp(int,int,Object):void
provisionalAckCfm(SipEndUser,int,Serializable,String):void
startReservationInd(int,int,Object):void
startReservationInd(int,int,Object):void

FIG 38
(Part 2)


3800
OfferorServiceUser
OfferorListener
702
OfferorServiceProvider
Provider
704
OfferorSIPServiceUser
SipUserAgentClientListener
3702
OfferorSIPServiceProvider
SipUserAgent
3704
AnswererSIPServiceProvider
SipUserAgent
3706
AnswererSIPServiceUser
SipUserAgentServerListener
3708
AnswererServiceProvider
Provider
706
AnswererServiceUser
AnswererListener
708
Coordination of startReservationRsp
Case: Offerer requests coordination before the Answerer issues the startReservationRsp
Case:successful-no adjusted contract at either side.
The 180 Tying is sent when this synchronization successfully occurs
startReservationRsp(int,int,Object):void
reservationCoordReq(SipEndUser,int,Serializable,String):void
reservationCoordInd(SipEndUser,int,String,Serializable,String):void
reservationCoordRsp(SipEndUser,int,Serializable,String):void
reservationCoordCfm(SipEndUser,int,String,Serializable,String):void
connectionStatusReq(SipEndUser,int,int,boolean,Serializable,String):void - 180 Alerting
startReservationRsp(int,int,Object):void
alertInd(int,int):void
connectionStatusInd(SipEndUser,int,int,String,Serializable,String):void-180 Alerting
provisionalAckReq(SipEndUser,int,Serializable,String):void
provisionalAckInd(SipEndUser,int,String,Serializable,String):void
provisionalAckRsp(SipEndUser,int,Serializable,String):void
provisionalAckCfm(SipEndUser,int,String,Serializable,String):void
negotiationRsp(int,int,Object):void
connectionRsp(SipEndUser,int,int,Serializable,String):void
connectionCfm(SipEndUser,int,String,Serializable,String):void
connectionStatusInd(SipEndUser,int,int,String,String,Serializable,String):void - ACK
completionInd(int,int,Object):void
negotiationCfm(int,int,Object):void

TAB. 1

3900

public void **bindReq**(OffererServiceUser oSu, AnswererServiceUser a Su, int spId)
Binds the given Service User's event listeners to the given E2ENP UA.
public void **registration Req**(int spId, int serviceUserId, java.lang.String user, java.lang.Object info)
Registers a user with the UA and optionally with a remote registrar.
public void **preNegotiationReq**(java.lang.String from, java.lang.String to, int serviceUserId, java.lang.Object offer)
Initiates a pre-negotiation phase.
public void **preNegotiationRsp**(int serviceProviderId, java.lang.Object answer)
Answers to an incoming request from the peer for pre-negotiation.
public void **renewLeaseReq**(java.lang.String user, int serviceUserId, java.lang.Object offer)
Initiates a pre-negotiation information lease refresh.
public void **renewLeaseRsp**(int serviceProviderId, java.lang.Object answer)
Answers to an incoming request from the peer for pre-negotiation information lease refresh.
public void **bookNegotiationReq**(java.lang.String user, int serviceUserId, int priority)
Before initiating the negotiation phase, this primitive allows the Service User to book the mutually exclusive use of the resource reservation process, with respect to concurrent instances of Service Users.
public void **negotiationReq**(java.lang.String from, java.lang.String to, int serviceUserId, int serviceProviderId, java.lang.object offer)
Initiates a negotiation phase.
public void **negotiationStatusRsp**(int serviceProviderId, java.lang.Object message)
Generates an intermediate signaling during negotiation in respose to a peer's signaling.
public void **bookReNegotiationReq**(int serviceUserId, int priority)
Before initiating the re-negotiation phase, this primitive allows the Service User to book the mutually exclusive use of the resource reservation process, with respect to concurrent instances of Service Users. In this case, the serviceProviderId must be specified.
public void **reNegotiationReq**(int serviceProviderId, boolean isPlanned, java.lang.Object offer)
Initiates a re-negotiation phase.
public void **reNegotiationRsp**(int serviceProviderId, java.lang.Object answer)
Answers to an incoming request from the peer for re-negotiation.
public void **reNegotiationStatusRsp**(int serviceProviderId, java.lang.Object message)
Generates an intermediate signaling during re-negotiation in response to a peer's signaling.
public void **startReservationRsp**(int serviceProviderId, java.lang.Object result)
Notifies that the Service User has completed both local and network resource reservation.
public void **releaseReq**(int serviceProviderId, java.lang.Object message)
Release the given phase (during pre-negotiation and lease renewal) or the overall session.
public void **releaseRsp**(int serviceProviderId, java.lang.Object message)
Replies to a request from the peer to release the given phase (during pre-negotiation and lease renewal) or the overall session.
public void **resetReq**(int serviceUserId, int serviceProviderId)
Resets the given session as an emergency procedure.
public void **resetRsp**(int serviceUserId, int serviceProviderId)
Replies to a request from the peer to reset the given session as an emergency

TAB. 2
(Part 1)

4000

public void **bindCfm** (Boolean result)
Returns the result of the bindReq primitive.
public void **registrationCfm** (int serviceUserId, java.lang.String user, java.lang.Object info)
Confirms the registration of the given user with the UA and optionally with an external
registrar.
public void **preNegotiationInd**(java.lang.String from, java.lang.String to, int serviceProviderId, java.lang.Object offer)
Notifies the Service User of a request from the peer to initiate a pre-negotiation phase.
public void **preNegotiationCfm**(int serviceUserId, int serviceProviderId, java.lang.Object answer)
Notifies the Service User of a reply from the peer concerning the given pre-negotiation phase.
public void **renewLeaseInd**(int serviceUserId, int serviceProviderId, java.lang.Object offer)
Notifies the Service User of a request from the peer to initiate a pre-negotiation information lease refresh.
public void **renewLeaseCfm**(int serviceUserId, int ServiceProviderId, java.lang.Object answer)
Notifies the Service User of a reply from the peer concerning the given pre-negotiation information lease refresh.
public void **negotiationStatusInd**(int ServiceUserId, int serviceProviderId, java.lang.Object message)
Notifies the Service User of an intermediate signaling during negotiation, in correspondence to a peer's signaling.
public void **bookReNegotiationCfm**(int serviceUserId, int serviceUserId, boolean isSuccessful)
Before initiating the re-negotiation phase, this primitive notifies the Service User that the E2ENP UA has successfully booked the mutually exclusive use of the resource reservation process, with respect to concurrent instances of Service Users.
public void **reNegotiationInd**(int serviceUserId, int serviceProviderId, boolean isPlanned, java.lang.Object offer)
Notifies the Service User of a request from the peer to initiate a re-negotiation phase.
public void **reNegotiationCfm**(int serviceUserId, int serviceProviderId, java.lang.Object answer)
Notifies the Service User of a reply from the peer concerning the given re-negotiaton.
public void **reNegotiationStatusInd**(int serviceUserId, int serviceProviderId, java.lang.Object message)
Notifies the Service User of an intermediate signaling during re-negotiation, in correspondence to a peer's signaling.
public void **startReservationInd**(int serviceUserId, int serviceProviderId, java.lang.Object result)

TAB. 2 (Part 2)

4000

public void **releaseCfm**(int serviceUserId, int serviceProviderId, java.lang.Object message)
Notifies the Service User of a reply from the peer concerning the release of the given phase (during pre-negotiation and lease renewal) or the overall session.
public void **alertInd**(int serviceUserId, int serviceProviderId)
Notifies the Service User that a remote peer has called in.
public void **ringBackInd**(int serviceUserId, int serviceProviderId)
Notifies the Service User that the remote peer has been alerted.
public void **congestionInd**(int serviceUserId, int serviceProviderId)
Notifies the given Service User instance that other higher priority instances are waiting for the mutex to be released. The Service User shall pre-empt by rolling back to the state before the negotiation/re-negotiation started, and invoke the release primitive.
public void **failureInd**(int serviceUserId, int serviceProviderId)
Notifies the Service User that an E2ENP UA internal error occurred and that the Service User shall invoke the release primitive.
public void **abortInd**(int serviceUserId, int serviceProviderId, int reason)
Notifies the Service User that a SIP error occurred and that the Service User shall simply consider the current operation as aborted. A reason is passed as well, indicating one of the possible sources of error:
0 - redirection,
1 - client error,
2 - server error,
3 - network failure.
public void **resetInd**(int serviceUserId, int serviceProviderId)
Notifies the Service User of a request from the peer concerning the reset of the session.
public void **resetCfm**(int serviceUserId, int serviceProviderId)

TAB. 3

4100

public void **bookNegotiationCfm**(int serviceUserId, int serviceUserId, boolean isSuccessful) Before initiating the negotiation phase, this primitive notifies the Service User that the E2ENP UA has successfully booked the mutually exclusive use of the resource reservation process, with respect to concurrent instances of Service Users. This primitive is specific of the offerer side, whereas the homonymous bookReNegotiationCfm may be invoked either at the offerer or at the answerer side. public void **negotiationCfm**(int serviceUserId, int serviceProviderId, java.lang.Object Answer) Notifies the Service User of a reply from the peer concerning the given negotiation.

TAB. 4

4200

public void **negotiationInd**(java.lang.String from, java.lang.String to, int serviceUserId, int serviceProviderId, java.lang.Object offer) Notifies the Service User of a request from the peer to initiate a negotiation phase. public void **completionInd**(nt serviceUserId, int serviceProviderId, Object message) Notifies the Service User that the negotiation/setup phase has been completed successfully.

TAB. 5

4300

public LevelOneEntry	**LevelOneCache.createEntry**(CoreSession session,int serviceProviderId) Creates a level 1 entry.
public LevelOneEntry	**LevelOneCache.getEntry**(int serviceProviderId) Gets a level 1 entry by using the service provieder ID as primary key.
public LevelOneEntry	**LevelOneCache.findEntry**(int serviceUserId) Gets a level 1 entry using the service user ID as secondary key.
public LevelOneEntry	**LevelOneCache.findEntry**(java.lang.String from,java.lang.String to) Gets a level 1 entry using the(From,To)SIP address couple as secondary key.
public java.util.Vector	**LevelOneCache.findEntryByExtSessionId**(java.lang.String sessionId) Gets the level 1 entries by using the E2ENP Session ID as secondary key.
public void	**LevelOneCache.removeEntry**(LevelOneEntry entry) Deletes the given entry from the cache.
public void	**LevelOneCache.clear()** Removes all the entries.
public void	**LevelOneEntry.putSessionID**(java.lang.String sessionId) Puts the E2ENP Session ID in the entry associated with the given service provider ID.
public java.lang.String	**LevelOneEntry.getSessionID()** Gets the E2ENP Session ID associated with the given service provider ID.
public void	**LevelOneEntry.putServiceUserID**(int serviceUserId) Puts the service user ID in the entry associated with the given service provider
public int	**LevelOneEntry.getServiceUserID()** Gets a service user ID.
public int	**LevelOneEntry.getServiceProviderID()** Gets a service provider ID.
public void	**LevelOneEntry.putFromAddress**(java.lang.String from) Puts the SIP From Address associated with the given service provider ID.
public java.lang.String	**LevelOneEntry.getFromAddress()** Gets the SIP From Address associated with the given service provider ID.
public void	**LevelOneEntry.putToAddress**(java.lang.String to) Puts the SIP To Address associated with the given service provider ID.
public java.lang.String	**LevelOneEntry.getToAddress()** Gets the SIP To Address associated with the given service provider ID.

TAB. 6

4400

public LevelTwoEntry	**LevelTwoCache.createEntry**(int serviceProviderId) Creates a level 2 entry.
public LevelTwoEntry	**LevelTwoCache.findEntry**(java.lang.String sessionId) Gets all the level 2 entries by using the E2ENP Session ID as primary key.
public void	**LevelTwoCache.removeEntry**(LevelTwoEntry entry) Deletes the given entry from the cache.
public void	**LevelTwoCache.clear()** Removes all the entries.
public void	**LevelTwoEntry.putSessionID**(java.lang.String sessionId) Puts the E2ENP Session ID in the entry associated with the given service provider ID.
public java.lang.String	**LevelTwoEntry.getSessionID()** Gets the E2ENP Session ID associated with the given service provider ID.
public void	**LevelTwoEntry.putServiceUserID**(int serviceUserId) Puts the service user ID in the entry associated with the given service provider
public int	**LevelTwoEntry.getServiceUserID()** Gets a service user ID.

TAB. 7
(Part 1)

4500

java.lang.String	**getType()** Returns an identifier to uniquely specifiy the type of this parser.
boolean	**isConfirmExpected**(java.io.Serializable in) Checks all media stream definitions for preconditions (PC), and, if at least one such definition has a PC that the offerer is going to carry out reservation, and thus, going to send a confirmation to the answerer when done.
boolean	**isKeepResultsCached**(java.io.Serializable in) Extracts information about whether the results of a negotiation of re-negotiations should be cached or not.
boolean	**isRingBackExpected**(java.io.Serializable in) Checks all media stream definitions and returns true, if any of the streams is going to be generated by the answerer.
java.lang.Object	**parseFinalResponse**(java.io.Serializable in) Parses a final result message and returns the corresponding status.
java.lang.Object	**parseLeaseAnswer**(java.io.Serializable in) Parses a transport representation of a renew lease answer and returns the corresponding object representation.
java.lang.Object	**parseLeaseOffer**(java.io.Serializable in) Parses a transport representation of a renew lease offer and returns the corresponding object representation.
long	**parseLeaseTime**(java.io.Serializable in) Parses a transport representation of a lease answer/offer and returns the corresponding lease time as a basic long value.
java.util.Vector	**parseListOfUsedSessionIds**(java.io.Serializable in) Parses the given input selectively to extract the external representation of the list of referenced E2ENP session identifier.
java.lang.Object	**parseMessage**(java.io.Serializable in) Parses content which is mapped over various e2enp primitives/SIP messages.
java.lang.Object	**parseNegAnswer**(java.io.Serializable in) Parses a transport representation of a negotiation answer and returns the corresponding object representation.
java.lang.Object	**parseNegCOffer**(java.io.Serializable in) Parses a transport representation of a negotiation counter-offer and returns the corresponding object representation.
java.lang.Object	**parseNegOffer**(java.io.Serializable in) Parses a transport representation of a negotiation offer and returns the corresponding object representation.

TAB. 7 (Part 2)

4500

java.lang.Object	**parsePreNegAnswer**(java.io.Serializable in) Parses a transport representation of a pre-negotiation answer and returns the corresponding object representation.
java.lang.Object	**parsePreNegOffer**(java.io.Serializable in) Parses a transport representation of a pre-negotiation offer and returns the corresponding object representation.
java.lang.Object	**parseReNegAnswer**(java.io.Serializable in) Parses a transport representation of a re-negotiation answer and returns the corresponding object representation.
java.lang.Object	**parseReNegCOffer**(java.io.Serializable in) Parses a transport representation of a re-negotiation counter-offer and returns the corresponding object representation.
java.lang.Object	**parseReNegOffer**(java.io.Serializable in) Parses a transport representation of a re-negotiation offer and returns the corresponding object representation.
java.lang.Object	**parseReservationResult**(java.io.Serializable in) Parses a transport representation of a reservation result message and returns the corresponding object representation.
java.lang.String	**parseSessionId**(java.io.Serializable in) Parses the given input selectively to extract the external representation of the E2ENP session identifier.

TAB. 8

4600

Parser	**createParser**(java.lang.String protocolID) Creates a parser instance for the given protocol.
void	**registerParser**(java.lang.String protocolID, java.lang.String parserClassName) Registers a parser implementation for the given protocol.

TAB. 9

4700

java.io.Serializable	**createFinalResponse**(java.lang.Object status) Creates the transport representation for a final response message.
java.io.Serializable	**createLeaseAnswer**(java.lang.Object in) Creates the transport representation for a renew lease answer.
java.io.Serializable	**createLeaseOffer**(java.lang.Object in) Creates the transport representation for a renew lease offer.
java.io.Serializable	**createMessage**(java.lang.Object in) Creates serializable representation of content which is mapped over various
java.io.Serializable	**createNegAnswer**(java.lang.Object in) Creates the transport representation for a negotiation answer.
java.io.Serializable	**createNegCOffer**(java.lang.Object in) Creates the transport representation for a negotiation counteroffer.
java.io.Serializable	**createNegOffer**(java.lang.Object in) Creates the transport representation for a negotiation offer.
java.io.Serializable	**createPreNegAnswer**(java.lang.Object in) Creates the transport representation for a pre-negotiation answer.
java.io.Serializable	**createPreNegOffer**(java.lang.Object in) Creates the transport representation for a pre-negotiation offer.
java.io.Serializable	**createReNegAnswer**(java.lang.Object in) Creates the transport representation for a re-negotiation answer.
java.io.Serializable	**createReNegCOffer**(java.lang.Object in) Creates the transport representation for a re-negotiation counteroffer.
java.io.Serializable	**createReNegOffer**(java.lang.Object in) Creates the transport representation for a re-negotiation offer.
java.io.Serializable	**createReservationResult**(java.lang.Object in) Creates the transport representation for a reservation result message.
long	**getLeaseTime**(java.lang.Object in) Extracts the lease time stored in the internal representation represented by the input parameter 'in'.
java.util.Vector	**getListOfUsedSessionIds**(java.lang.Object in) Extracts the list of used session IDs stored in the internal representation represented by the input parameter 'in'.
java.lang.String	**getSessionId**(java.lang.Object in) Extracts the session ID stored in the internal representation represented by the input parameter 'in'.
java.lang.String	**getType()** Returns an identifier to uniquely specifiy the type of this factory.
boolean	**isKeepResultsCached**(java.lang.Object in) Extracts information about whether the results of a negotiation or of re-negotiations

TAB. 10

4800

Factory	**createFactory**(java.lang.String protocolID)
void	**registerFactory**(java.lang.String protocolID, java.lang.String factoryClassName) Registers a factory class for the given protocol.

TAB. 11

4900

abstract Factory	**createFactory**(java.lang.String protocolID) Creates a factory instance for the given protocol.
abstract Parser	**createParser**(java.lang.String protocolID) Creates a parser instance for the given protocol.
static E2ENPContentHandlerFactory	**getInstance**() Provides access to the singleton instance of the E2ENPContentHandlerFactory.
abstract void	**registerFactory**(java.lang.String protocolID, java.lang.String factoryClassName) Registers a factory class for the given protocol.
abstract void	**registerImplementation**(java.lang.String ID, java.lang.String parserClassName, java.lang.String factoryClassName) Registers a parser and factory implementation using the given ID.
abstract void	**registerParser**(java.lang.String protocolID, java.lang.String parserClassName) Registers a parser implementation for the given protocol.

TAB. 12

5000

void **release**()
Shuts down the given SIP UA.
void **bindReq**(SipEndUser user, SipUserAgentClientListener ssmIc, SipUserAgentServerListener ssmIs)
Binds the given user's event listeners to the given SIP UA.
void **configureReq**(SipManagementListener sml, SipConfigurationReq conf)
Configures the given SIP UA.
void **connectionReq**(SipEndUser user, int connectionId, SipConnectionReq invitation, java.io.Serializable message, java.lang.String mimeContentType)
Allows generating a request to establish a session.
void **connectionRsp**(SipEndUser user, int connectionId, int status, java.io.Serializable message, java.lang.String mimeContentType)
Allows replying to a request to establish a session.
void **connectionStatusReq**(SipEndUser user, int connectionId, int type, boolean isMiddle, java.io.Serializable message, java.lang.String mimeContentType)
Allows generating provisional responses.
void **connectionStatusRsp**(SipEndUser user, int connectionId, int type, java.io.Serializable message, java.lang.String mimeContentType)
Allows generating explicitly an ACK.
void **disconnectionReq**(SipEndUser user, int connectionId, java.io.Serializable message, java.lang.String mimeContentType)
Allows closing a session.
void **disconnectionRsp**(SipEndUser user, int connectionId)
Allows replying to an incoming request to close a session.
void **optionsReq**(SipEndUser user, int connectionId, java.lang.String target, java.io.Serializable body, java.lang.String mimeContentType)
Allows sending OPTIONS.
void **optionsRsp**(SipEndUser user, int connectionId, java.io.Serializable body, java.lang.String mimeContentType)
Allows replying to an incoming OPTIONS.
void **messageReq**(SipEndUser user, int connectionId, java.lang.String target, java.io.Serializable body, java.lang.String mimeContentType)
Allows sending a MESSAGE.
void **messageRsp**(SipEndUser user, int connectionId, java.io.Serializable body, java.lang.String mimeContentType)
Allows replying to an incoming MESSAGE.
void **provisionalAckReq**(SipEndUser user, int connectionId, java.io.Serializable message, java.lang.String mimeContentType)
Allows sending a PRACK.
void **provisionalAckRsp**(SipEndUser user, int connectionId, java.io.Serializable message, java.lang.String mimeContentType)
Allows replying to an incoming PRACK.
void **registerReq**(SipEndUser user, SipRegistrationReq registration, java.io.Serializable body, java.lang.String bodytype)
Allows registering the specified user at the given SIP Registrar.
void **reservationCoordReq**(SipEndUser user, int connectionId, java.io.Serializable message, java.lang.String mimeContentType)
Allows sending an UPDATE.
void **reservationCoordRsp**(SipEndUser user, int connectionId, java.io.Serializable message, java.lang.String mimeContentType)
Allows replying to an incoming UPDATE.

TAB. 13

5100

void **connectionCfm**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a SIP session establishment.
void **provisionalAckCfm**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a PRACK.
void **registerCfm**(SipEndUser user, SipRegistrationCfm registration, int status, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a registration request.

TAB. 14

5200

void **connectionInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Indicates an incoming request of SIP session establishment.
void **provisionalAckInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Indicates an incoming PRACK.

TAB. 15

5300

void **connectionStatusInd**(SipEndUser user, int connectionId, int type, java.lang.String typeString, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Indicates an incoming provisional response from the peer.
void **disconnectionCfm**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a former request to close the session.
void **disconnectionInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Indicates a request from the peer to close the session.
void **optionsInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Notifies the incoming of OPTIONS.
void **messageInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Notifies the incoming of a MESSAGE.
void **optionsCfm**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a former request to send OPTIONS.
void **messageCfm**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a former request to send a MESSAGE.
void **reservationCoordCfm**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a former request to send an UPDATE.
void **reservationCoordInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Indicates an incoming UPDATE sent by the peer.

TAB. 16

5400

void **bindReq**(SipEndUser user, SipRegistrarListener ssml) Binds the given user's event listener to the given SIP UA. void **configureReq**(SipManagementListener sml, SipConfigurationReq conf) Configures the given SIP UA. void **optionsRsp**(SipEndUser user, int connectionId, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges OPTIONS. void **messageRsp**(SipEndUser user, int connectionId, java.io.Serializable body, java.lang.String mimeContentType) Acknowledges a MESSAGE. void **registerRsp**(SipEndUser user, int connectionId, SipRegistrationRsp registration, int status, java.io.Serializable body, java.lang.String bodytype) Primitive allowing a Registrar to reply to a request for registration from SIP users.

TAB. 17

5500

void **registerInd**(SipEndUser user, int connectionId, SipRegistrationInd registration, java.io.Serializable body, java.lang.StringEmimeContentType) Indicates a request from the peer to make a registration. void **optionsInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Notifies the incoming of OPTIONS. void **messageInd**(SipEndUser user, int connectionId, java.lang.String from, java.io.Serializable body, java.lang.String mimeContentType) Notifies the incoming of a MESSAGE.

TAB. 18
(Part 1)

5600

Source State	Triggering event	Guard condition	Action	Target state
Root.Idle	disconnectionCfm		releaseCfm	Root.Idle
"	releaseRsp		disconnectionRsp	Root.Idle
Root.WaitRenewLeaseRsp	Timer T2 expires		failureInd	Root.WaitToClear
Root.WaitRenewLeaseCfm	connectionStatusInd	Status >=300	abortInd	Root.Idle
"	Timer T101 expires		failureInd	Root.WaitToClear
Root.WaitPreNegotRsp	Timer T1 expires		failureInd	Root.WaitToClear
Root.WaitPreNegotCfm	connectionStatusInd	Status >=300	abortInd	Root.Idle
"	Timer T102 expires		failureInd	Root.WaitToClear
Root.NegOfferer	Timer T103 expires		failureInd	Root.WaitToClear
Root.ReNegOfferer.WaitNegReq	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
"	Timer T5 expires		_ResvMtx.signal abortInd	Root.Idle
Root.NegOfferer.WaitTryingInd	connectionStatusInd	Status >=300	abortInd	Root.Idle
Root.NegOfferer.WaitProgressInd	connectionStatusInd	Status >=300	abortInd	Root.Idle
Root.NegOfferer.WaitNegStatusRsp	Timer T6 expires		failureInd	Root.WaitToClear
Root.NegOfferer.WaitPrackCfm	connectionStatusInd	Status >=300	failureInd	Root.WaitToClear
Root.NegOfferer.WaitReservCompl	Timer T7 expires		failureInd	Root.WaitToClear
Root.NegOfferer.WaitRCoordCfm	connectionStatusInd	Status >=300	failureInd	Root.WaitToClear
Root.NegOfferer.WaitRinging	connectionStatusInd	Status >=300	failureInd	Root.WaitToClear

TAB. 18 (Part 2)

5600

Source State	Triggering event	Guard condition	Action	Target state
Root.NegOfferer.WaitFinalProvAckCfm	connectionStatusInd	Status >=300	failureInd	Root.WaitToClear
Root.NegOfferer.WaitConnectionCfm	connectionStatusInd	Status >=300	failureInd	Root.WaitToClear
Root.NegOfferer.WaitContractAdjusted	Timer T8 expires		failureInd	Root.WaitToClear
Root.ReNegOfferer	Timer T104 expires		failureInd	Root.WaitToClear
Root.ReNegOfferer.WaitReNegReq	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
"	Timer T9 expires		_ResvMtx.signal abortInd	Root.Idle
Root.ReNegOfferer.WaitTryingInd	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
Root.ReNegOfferer.WaitProgressInd	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
Root.ReNegOfferer.WaitNegStatusRsp	Timer T10 expires		failureInd	Root.WaitToClear
Root.ReNegOfferer.WaitPrackCfm	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
Root.ReNegOfferer.WaitReservCompl	Timer T11 expires		failureInd	Root.WaitToClear
Root.ReNegOfferer.WaitCoordCfm	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
Root.ReNegOfferer.WaitRinging	connectionStatusInd	Status >=300	failureInd	Root.WaitToClear
Root.ReNegOfferer.WaitFinalProvAckCfm	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
Root.ReNegOfferer.WaitFConnectionCfm	connectionStatusInd	Status >=300	abortInd	Root.SessionEstablished
Root.ReNegOfferer.WaitContractAdjusted	Timer T12 expires		failureInd	Root.WaitToClear
Root.NegAnswerer	negotiationReq		abortInd	Root.NegAnswerer
"	reNegotiationReq		abortInd	Root.NegAnswerer

TAB. 18 (Part 3)

5600

Source State	Triggering event	Guard condition	Action	Target state
"	Timer T105 expires		failureInd	Root.WaitToClear
Root.NegAnswerer.WaitAnswer	Timer T13 expires		failureInd	Root.WaitToClear
Root.NegAnswerer.WaitCounterOffer				
Root.NegAnswerer.WaitNegStatusRsp	Timer T14 expires		failureInd	Root.WaitToClear
Root.NegAnswerer.WaitFinalProvAck				
Root.NegAnswerer.WaitFinalAnswer	Timer T15 expires		failureInd	Root.WaitToClear
Root.NegAnswerer.HoldOnAnswer				
Root.NegAnswerer.WaitConnectionCfm				
Root.NegAnswerer.WaitForCoordDone.WaitLocalCoord	Timer T19 expires		failureInd	Root.WaitToClear
Root.NegAnswerer.WaitForCoordDone.WaitRemoteCoord	Timer T20 expires		failureInd	Root.WaitToClear
Root.NegAnswerer.WaitForCoordDone.LocalTokenAchieved				
Root.NegAnswerer.WaitForCoordDone.RemoteTokenAchieved				
Root.ReNegAnswerer	Timer T106 expires		failureInd	Root.WaitToClear
Root.ReNegAnswerer.WaitAnswer	Timer T16 expires		failureInd	Root.WaitToClear
Root.ReNegAnswerer.WaitCounterOffer				
Root.ReNegAnswerer.WaitNegStatusRsp	Timer T17 expires		failureInd	Root.WaitToClear

TAB. 18 (Part 4)

5600

Source State	Triggering event	Guard condition	Action	Target state
Root.ReNegAnswerer.WaitFinalProvAck				
Root.ReNegAnswerer.WaitFinalAnswer	Timer T18 expires		failureInd	Root.WaitToClear
Root.ReNegAnswerer.HoldOnAnswer				
Root.ReNegAnswerer.WaitConnectionCfm				
Root.ReNegAnswerer.WaitForCoordDone.WaitLocalCoord	Timer T21 expires		failureInd	Root.WaitToClear
Root.ReNegAnswerer.WaitForCoordDone.WaitRemoteCoord	Timer T22 expires		failureInd	Root.WaitToClear
Root.ReNegAnswerer.WaitForCoordDone.LocalTokenAchieved				
Root.ReNegAnswerer.WaitForCoordDone.RemoteTokenAchieved				
Root.SessionEstablished				
Root.WaitReleaseRsp	disconnectionCfm		releaseCfm	Root.WaitReleaseRsp
"	Timer T3 expires		releaseReq	Root.Wait DisconnectionCfm
Root.WaitToClear	Timer T4 expires		releaseReq	Root.Wait DisconnectionCfm
Root.WaitDisconnectionCfm	releaseRsp		disconnectionRsp	Root.Wait DisconnectionCfm

TAB. 19 (Part 1)

5700

Name	Description	Default Value
T1	Waiting pre-negotiation response from E2ENP UA API Service User	TBD
T2	Waiting lease renew response from E2ENP UA API Service User	TBD
T3	Waiting release response from E2ENP UA API Service User	TBD
T4	Waiting release request from E2ENP UA API Service User, after the E2ENP UA has generated a failure indication or a congestion indication to the Service User	TBD
T5	Waiting negotiation request from E2ENP UA API Service User after having booked the resource reservation mutex at the offerer side	TBD
T6	Waiting negotiation status response with counter-offer, from E2ENP UA API Service User at the offerer side	TBD
T7	Waiting start reservation response from E2ENP UA API Service User at the offerer side	TBD
T8	Waiting negotiation status response from E2ENP UA API Service User to handle contract adjustment at the offerer side	TBD
T9	Waiting re-negotiation request from E2ENP UA API Service User after having booked the resource reservation mutex at the offerer side	TBD
T10	Waiting re-negotiation status response with counteroffer, from E2ENP UA API Service User at the offerer side	TBD
T11	Waiting start reservation response from E2ENP UA API Service User, during re-negotiation at the offerer side	TBD
T12	Waiting re-negotiation status response from E2ENP UA API Service User to handle contract adjustment at the offerer side	TBD
T13	Waiting negotiation status response with answer from E2ENP UA API Service User at the answerer side	TBD
T14	Waiting negotiation status response with counteroffer from E2ENP UA API Service User at the answerer side	TBD
T15	Waiting negotiation response from E2ENP UA API Service User at the answerer side	TBD
T16	Waiting re-negotiation status response with answer from E2ENP UA API Service User at the answerer side	TBD

TAB. 19 (Part 2)

5700

Name	Description	Default Value
T17	Waiting re-negotiation status response with counteroffer from E2ENP UA API Service User at the answerer side	TBD
T18	Waiting re-negotiation response from E2ENP UA API Service User at the answerer side	TBD
T19	Waiting start reservation response from E2ENP UA API Service User at the offerer side during negotiations	TBD
T20	Waiting negotiation response from E2ENP UA API Service User side for handling adjusted contracts during negotiations	TBD
T21	Waiting start reservation response from E2ENP UA API Service User at the offerer side during re-negotiations	TBD
T22	Waiting negotiation response from E2ENP UA API Service User side for handling adjusted contracts during re-negotiations	TBD
T101	Waiting OPTIONS confirmation from SIP API for handling lease renewal primitive	TBD
T102	Waiting MESSAGE confirmation from SIP UA for handling pre-negotiation primitive	TBD
T103	Exceeding delay in receiving any SIP UA Generic API primitive while acting as offerer in a negotiation process	TBD
T104	Exceeding delay in receiving any SIP UA Generic API primitive while acting as answerer in a negotiation process	TBD
T105	Exceeding delay in receiving any SIP UA Generic API primitive while acting as offerer in a re-negotiation process	TBD
T106	Exceeding delay in receiving any SIP UA Generic API primitive while acting as answerer in a re-negotiation process	TBD